



Fusão sensorial e cooperação em equipas de robôs móveis

Pedro Miguel da Silva Rocha Relvas

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor António Paulo Gomes Mendes Moreira

Co-orientador: Professor Doutor Paulo José Cerqueira Gomes da Costa

12 de Julho de 2017

Resumo

Nos últimos anos, tem-se verificado uma crescente aposta na utilização de soluções robóticas para resolver problemas nas mais diversas áreas de aplicação.

As soluções multi-agente permitem obter resultados de uma forma mais rápida, eficiente e, por vezes, com menos custos do que a utilização de um único agente. As áreas de aplicação vão desde a busca e salvamento, segurança e vigilância, mapeamento, entre outros. Para garantir o sucesso e maximizar o desempenho dos agentes durante uma missão, deverão existir mecanismos que garantam a coordenação e cooperação entre agentes. Na verdade, não há garantias que todos os agentes consigam observar todos os objetos de interesse e, para além disso, existem erros e ruído na sensorização do ambiente, que pode prejudicar o cumprimento da missão. Desta forma, a informação percecionada deve ser partilhada e fundida pelos agentes, permitindo construir uma representação do mundo mais robusta e completa. Em certas aplicações, o seguimento de alvos é igualmente um requisito fundamental, podendo definir o sucesso ou insucesso da missão.

A presente dissertação apresenta um supervisor capaz de coordenar e fundir a informação de um conjunto de agentes num ambiente dinâmico, garantindo a correta cooperação e ação perante as mais diversas situações a que estes estão sujeitos, mesmo que o número de agentes operacionais varie ao longo do tempo. Como área de aplicação, utilizaram-se os robôs da equipa de futebol robótico 5DPO. Para além disso, foram desenvolvidas e comparadas algumas abordagens de estimação de posição e velocidade de um alvo (nomeadamente, a bola de jogo) num referencial móvel, por parte de cada agente, assim como algumas abordagens de fusão da mesma, por parte do supervisor. Estas abordagens baseiam-se em filtros de Kalman e outros métodos mais heurísticos.

Abstract

In the past few years, there has been a growing usage of robotic solutions to solve problems in a wide range of application areas.

Multi-agent solutions can reach results in a much faster and efficient way and, sometimes, with a lower cost than a single-agent application. Applications areas range from search and rescue, security and surveillance, mapping, among others. To ensure the success and to maximize the agent performance during a mission, some mechanisms must be in place to ensure the coordination and cooperation between the agents. In fact, there is no guarantee that all the agents will be able to observe all the objects of interest and, besides that, there are some errors and observation noise in the sensing of the environment, which can affect the fulfillment of the mission. Hence, the perceived information must be shared and merged by the agents, allowing to build a more robust and complete worldstate. In certain applications, object tracking is also a requirement that can define the success or not of the mission.

This dissertation presents a supervisor capable of coordinate and merge the information of a team of agents, in a highly dynamic environment. This supervisor guarantees the correct cooperation and actions in the most diverse situations to which the agents are subject to, even if the number of active agents varies throughout time. As a testbed, the robots of the robotic soccer team 5DPO were used. Besides that, some approaches to estimate the position and velocity of a target in a moving frame (each team agent), as well as some approaches to merge it in the supervisor. These approaches are based on Kalman filters and other more heuristic methods.

Agradecimentos

Antes demais, queria agradecer aos meus orientadores, Professor Doutor António Paulo Moreira e Professor Doutor Paulo Costa, por toda a disponibilidade, apoio, paciência e conhecimento transmitido aos longos destes últimos anos, com especial ênfase nestes últimos meses.

Um muito obrigado aos meus pais, irmã e avó, pelo apoio, incentivo, carinho demonstrados, e por me terem proporcionado esta oportunidade académica.

Aos meus amigos e colegas de curso com ênfase aos Filipes, Nuno, Faustino, Miguel Almeida, Joana, Alice, Gonçalo, Chico, Tony, Anselmo, Flávio, Rafa, Ricky, Pestana, Valdez, João, André e Nolasco, pela amizade, conhecimentos transmitidos e experiências vividas. Um abraço Sr. Vitor, pela disponibilidade, apoio e incentivo demonstrados. Ao Hélder, João e Ricardo, pelo apoio, compreensão e momentos de descontração proporcionados.

Agradeço também a todo o pessoal do CRIIS, OceanSys, colaboradores, membros e ex-membros da equipa 5DPO, em especial ao Andry, Héber, Filipe, Luís Rocha, Miguel, Tiago, Nima, Ivo, Fernando, João, Pedro, Carlos, Sr. Fernando, Jorge e Bruno.

Por fim, queria agradecer ao Professor Doutor Armando Sousa e ao Ricardo Dias, pela disponibilidade demonstrada, e a todos os professores que me foram guiando ao longo do meu percurso escolar e académico.

Pedro Relvas

*“Personally, I’m not afraid of a robot uprising.
The benefits far outweigh the threats.”*

Daniel H. Wilson

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	1
1.3	Motivação	2
1.4	Caracterização do Problema	2
1.4.1	Definição do Problema	2
1.4.2	Solução Proposta	3
1.5	Estrutura da Dissertação	3
2	Revisão Bibliográfica	5
2.1	Cooperação em Equipas de Robôs Móveis	5
2.1.1	Planeamento e controlo de tarefas	6
2.2	Fusão Sensorial e Seguimento de Alvos	7
2.3	Futebol Robótico	8
2.3.1	Robocup	8
2.3.2	Estratégias de Cooperação Entre Agentes	9
2.3.3	Estratégias de Fusão Sensorial	12
2.3.4	Estratégias de Estimação de Posição e Velocidade da Bola	14
2.4	Filtro de Kalman	17
2.4.1	Filtro de Kalman Estendido	19
2.4.2	<i>Unscented Kalman Filter</i>	20
3	Caracterização do Problema	23
3.1	Definição do Problema	23
3.2	Solução Proposta	24
4	Equipa 5DPO	25
4.1	Contexto	25
4.2	Hardware	25
4.3	Arquitetura do Software	26
4.3.1	Flashbus	26
4.3.2	Ovis	27
4.3.3	Coach	28
4.3.4	Decision	28
4.4	Arquitetura de Cooperação	28

5	Ambiente de Simulação	31
5.1	SimTwo	31
5.2	Simulador MSL 5DPO	32
5.2.1	Conclusões	33
6	Estimação de Posição e Velocidade da Bola por Parte de um Agente	37
6.1	Aquisição de medida da bola	38
6.2	Dinâmica da bola	39
6.3	Filtro de Kalman	42
6.4	Filtro de Kalman - Referencial Local	42
6.4.1	Dimensionamento do Filtro	45
6.4.2	Análise do Ruído de Medição da Bola	47
6.4.3	Ajuste da Covariância do Ruído de Medição da Bola	48
6.4.4	Análise do Ruído de Localização do Robô	50
6.4.5	Ensaio Realizados	52
6.4.5.1	Ensaio com Dados Gerados no Matlab	53
6.4.5.2	Ensaio com Dados Obtidos no Simulador	62
6.4.5.3	Ensaio com Dados Obtidos num Robô Real	68
6.5	Filtro de Kalman - Referencial Global	79
6.5.1	Dimensionamento do Filtro	80
6.5.2	Ensaio Realizados	83
6.5.2.1	Ensaio com Dados Gerados no Matlab	84
6.5.2.2	Ensaio com Dados Obtidos no Simulador	87
6.5.2.3	Ensaio com Dados Obtidos num Robô Real	91
6.5.2.4	Ensaio no <i>Software</i> de "Decisão" de cada robô	94
6.5.3	Inibição do Erro de Localização do Robô na Estimação da Velocidade da Bola	94
6.6	Estimação de Velocidade Através de uma Regressão Linear	96
6.6.1	Comparação de Estimação de Velocidade Através do Filtro de Kalman com Regressão Linear	97
6.7	Processo de Escolha de Bola no <i>Software</i> de Decisão	102
6.8	Conclusões	104
7	Treinador de equipa de Futebol Robótico	107
7.1	Novo Treinador da Equipa 5DPO	108
7.2	Arquitetura da Aplicação	109
7.3	<i>WorldState</i>	110
7.4	Ciclo Principal	110
7.4.1	Atualização do Estado da <i>RefereeBox</i>	112
7.4.2	Atualização da Tática	113
7.4.3	Atualização da Jogada	113
7.4.4	Papéis, Tarefas e Ações	113
7.5	Formações e Atualização de Papéis	114
7.6	Atribuição Dinâmica de <i>Roles</i>	116
7.7	Comunicações	119
7.7.1	ZeroMQ	120
7.7.2	Implementação	121
7.8	Conclusões	122

8	Fusão de Bola no Treinador	123
8.1	Abordagens de Fusão de Bola	123
8.2	Ensaio Realizado	124
8.3	Conclusões	144
9	Conclusões	147
9.1	Trabalho Realizado	147
9.2	Trabalho Futuro	149
A	Ensaio do Filtro de Kalman de Estimação de Posição e Velocidade da Bola no Software de Decisão da Equipa SDPO	151
B	Treinador	155
B.1	Diagrama UML das Classes e Enumerações Mais Relevantes do Treinador	156
B.2	Árvore de Decisão da Tática da Equipa	161
B.3	Formações e Máquinas de Estados das <i>Role Slots</i>	162
B.4	Fluxograma do algoritmo atribuição de <i>Slots</i> para todas as jogadas, à exceção da <i>playNormal</i>	164
B.5	Fluxograma do algoritmo atribuição de <i>Slots</i> para a jogada <i>playNormal</i>	166
	Referências	169

Lista de Figuras

2.1	Diagrama de operação do filtro de Kalman	18
2.2	Algoritmo de operação do filtro de Kalman	18
2.3	Dinâmica do Filtro de Kalman Estendido	19
2.4	Princípio da (<i>unscented transform</i>)	20
4.1	Robôs da equipa <i>Middle Size League</i> 5DPO	26
4.2	Arquitetura de <i>software</i> da equipa 5DPO	27
4.3	Imagem capturada pelo sistema de visão	27
4.4	Arquitetura Cooperação	29
5.1	Representação dos robôs da equipa MSL 5DPO no ambiente de simulação SimTwo	32
5.2	Fluxograma do ciclo principal do <i>script</i> de controlo do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo	34
5.3	Diagrama UML de classes do <i>script</i> de controlo do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo	35
5.4	Janela principal do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo	36
5.5	Janela <i>Sheets</i> do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo	36
6.1	Sistema de visão	38
6.2	Comparação de x_{medido} e $x_{estimado}$	41
6.3	Sistema de coordenadas da observação da bola	43
6.4	Sistema de coordenadas de observação da bola	47
6.5	Medidas de observação da bola por parte do robô 1	48
6.6	Regressão linear e desvio padrão da observação da bola	49
6.7	Posições de análise de ruído de localização dos robôs	50
6.8	Cenários de ensaio do filtro do estimador de posição e velocidade da bola	52
6.9	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - Posições da bola e do robô, no referencial global	55
6.10	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - Posição da bola no referencial local	55
6.11	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - velocidade da bola	55
6.12	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - Posições da bola e do robô, no referencial global	56

6.13	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, com ruído (Q1) - Posição da bola no referencial local	56
6.14	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - velocidade da bola	56
6.15	Comparação da velocidade estimada através da aproximação numérica da derivada da posição em função do tempo, utilizando as posições medidas ou utilizando as posições estimadas pelo filtro de Kalman	57
6.16	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, sem ruído (Q1) - posição da bola no referencial local	58
6.17	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - velocidade da bola	59
6.18	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - posição da bola no referencial local	59
6.19	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, com ruído (Q1) - velocidade da bola	59
6.20	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - posição da bola no referencial local	61
6.21	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, sem ruído (Q1) - velocidade da bola	61
6.22	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - posição da bola no referencial local	61
6.23	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - velocidade da bola	62
6.24	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - Posições da bola e do robô, no referencial global . . .	63
6.25	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - Posição da bola no referencial local	64
6.26	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - velocidade da bola	64
6.27	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos em SimTwo (Q1) - posição da bola no referencial local	65
6.28	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - velocidade da bola	65
6.29	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos em SimTwo (Q1) - posição da bola no referencial local	66
6.30	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - velocidade da bola	66
6.31	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos em SimTwo (Q2) - posição da bola no referencial local	67
6.32	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q2) - velocidade da bola	67
6.33	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - Posições da bola e do robô, no referencial global . .	70
6.34	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - Posição da bola no referencial local	70
6.35	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - velocidade da bola	70

6.36	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - Posição da bola no referencial local	71
6.37	Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola	71
6.38	Traçado das velocidades lineares do robô	71
6.39	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=0.9\text{rad/s}$, $d=1\text{m}$)	72
6.40	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=0.9\text{rad/s}$, $d=1\text{m}$)	73
6.41	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=0.9\text{rad/s}$, $d=2\text{m}$)	73
6.42	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=0.9\text{rad/s}$, $d=2\text{m}$)	73
6.43	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=1.8\text{rad/s}$, $d=1\text{m}$)	74
6.44	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=1.8\text{rad/s}$, $d=1\text{m}$)	74
6.45	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=1.8\text{rad/s}$, $d=2\text{m}$)	74
6.46	Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=1.8\text{rad/s}$, $d=2\text{m}$)	75
6.47	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q1) - posição da bola no referencial local	76
6.48	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - velocidade da bola	76
6.49	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local	77
6.50	Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola	77
6.51	Ensaio 4 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - Posições da bola e do robô, no referencial global	78
6.52	Ensaio 4 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local	78
6.53	Ensaio 4 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola	79
6.54	Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados gerados em Matlab, com ruído (Q1) - Posições da bola e do robô, no referencial global	85
6.55	Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados gerados em Matlab, com ruído (Q1) - Posição da bola no referencial local	85
6.56	Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados gerados em Matlab, com ruído (Q1) - velocidade da bola	86
6.57	Comparação da velocidade estimada através da aproximação numérica da derivada da posição em função do tempo, utilizando as posições medidas ou utilizando as posições estimadas pelo filtro de Kalman	86

6.58	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q1) - Posição da bola no referencial local	89
6.59	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q1) - velocidade da bola	89
6.60	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q2) - Posição da bola no referencial local	89
6.61	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q2) - velocidade da bola	90
6.62	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q3) - Posição da bola no referencial local	90
6.63	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q3) - velocidade da bola	90
6.64	Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos num robô real (Q2) - velocidade da bola	91
6.65	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos num robô real (Q3) - velocidade da bola	92
6.66	Ensaio 4 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos num robô real (Q3) - velocidade da bola	93
6.67	Vista de topo do cenário em que ocorre uma correção na localização de um agente, no referencial global	94
6.68	Velocidades da bola no referencial global no cenário de acontecer uma correção na localização do robô	95
6.69	Ensaio 1 - Posições medida e estimada da bola, no referencial local	98
6.70	Ensaio 1 - Velocidade estimada da bola em função do tempo, no referencial local	98
6.71	Ensaio 1 - Velocidade estimada da bola em função do tempo, no referencial global	99
6.72	Ensaio 2 - Posições medida e estimada da bola, no referencial local	99
6.73	Ensaio 2 - Velocidade estimada da bola em função do tempo, no referencial local	100
6.74	Ensaio 2 - Velocidade estimada da bola em função do tempo, no referencial global	100
6.75	Ensaio 3 - Posições medida e estimada da bola, no referencial local	101
6.76	Ensaio 3 - Velocidade estimada da bola em função do tempo, no referencial local	101
6.77	Ensaio 3 - Velocidade estimada da bola em função do tempo, no referencial global	101
6.78	Comparação entre velocidades estimadas, no referencial local, por diferentes abordagens	102
7.1	Novo treinador da equipa 5DPO	108
7.2	Diagrama UML de classes do treinador	110
7.3	Ciclo principal de controlo	111
7.4	Diagrama de atualização do estado da <i>Referee Box</i>	112
7.5	Diagrama de atualização da jogada da equipa	114
7.6	Formações e máquina de estados para a jogada <i>playBarrier</i>	116
7.7	Formações e máquina de estados para a jogada <i>playNormal</i>	117
7.8	Diagrama de Comunicações entre o treinador e o <i>software</i> de decisão	120
7.9	Teste 3, publicação de 400 mensagens de 8 bytes a 10 clientes	121
8.1	Ensaio 1 - cenário de ensaio - todos os agentes ativos	125
8.2	Ensaio 1 - qualidade das bolas	125
8.3	Representação no mapa	126
8.4	Ensaio 1 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes	126
8.5	Representação no mapa	126

8.6	Ensaio 1 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas	126
8.7	Representação no mapa	127
8.8	Ensaio 1 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade .	127
8.9	Representação no mapa	127
8.10	Ensaio 1 - Assunção de bola com maior qualidade	127
8.11	Ensaio 2 - cenário de ensaio - mudança repentina de posição da bola	129
8.12	Ensaio 2 - qualidade das bolas	129
8.13	Ensaio 2 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes	130
8.14	Ensaio 2 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas	131
8.15	Ensaio 2 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade .	132
8.16	Ensaio 2 - Assunção de bola com maior qualidade	133
8.17	Ensaio 3 - cenário de ensaio - falso positivo (duas bolas em campo)	135
8.18	Ensaio 3 - qualidade das bolas	135
8.19	Representação no mapa	136
8.20	Ensaio 3 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes	136
8.21	Representação no mapa	136
8.22	Ensaio 3 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas	136
8.23	Representação no mapa	137
8.24	Ensaio 3 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade .	137
8.25	Representação no mapa	137
8.26	Ensaio 3 - Assunção de bola com maior qualidade	137
8.27	Ensaio 4 - cenário de ensaio	139
8.28	Ensaio 4 - qualidade das bolas	139
8.29	Ensaio 4 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes	140
8.30	Ensaio 4 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas	141
8.31	Ensaio 4 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade .	142
8.32	Ensaio 4 - Assunção de bola com maior qualidade	143
A.1	Ensaio 1 do filtro de Kalman com vetor de estado no referencial global - implementação no <i>Decision</i> , com dados obtidos em SimTwo	151
A.2	Ensaio 2 do filtro de Kalman com vetor de estado no referencial global - implementação no <i>Decision</i> , com dados obtidos em SimTwo	152
A.3	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global - implementação no <i>Decision</i> , com dados obtidos em SimTwo	152
A.4	Ensaio 1 do filtro de Kalman com vetor estado no referencial global - implementação no <i>Decision</i> , com dados obtidos em ambiente real	153
A.5	Ensaio 2 do filtro de Kalman com vetor de estado no referencial global - implementação no <i>Decision</i> , com dados obtidos em ambiente real	153
A.6	Ensaio 3 do filtro de Kalman com vetor de estado no referencial global - implementação no <i>Decision</i> , com dados obtidos em ambiente real	154
B.2	Diagrama UML das Classes e Enumerações Mais Relevantes do Treinador	159
B.3	Diagrama de atualização da tática da equipa	161
B.4	Formações e máquina de estados para a jogada <i>playOurStart</i>	162
B.5	Formações e máquina de estados para a jogada <i>playSearchBall</i>	162

B.6	Formações e máquina de estados para a jogada <i>playDroppedBall</i>	163
B.7	Formações e máquina de estados para a jogada <i>playDefendPenalty</i>	163
B.8	Algoritmo de Atribuição de <i>slots</i> para todas as jogadas, à exceção da <i>playNormal</i>	165
B.9	Algoritmo de Atribuição de <i>slots</i> para a jogada <i>playNormal</i>	168

Lista de Tabelas

6.1	Desvio padrão das medições da bola	48
6.2	Desvio padrão e variância das medidas da pose do robô 1 para diferentes pontos .	51
6.3	Desvio padrão e variância das medidas da pose do robô 2 para diferentes pontos .	51
6.4	Desvio padrão e variância das medidas da pose do robô 3 para diferentes pontos .	51

Abreviaturas e Símbolos

2D	Espaço Bi-Dimensional
3D	Espaço Tri-Dimensional
AFLS	Adaptive Fuzzy Logic System
CAMBADA	Cooperative Autonomous Mobile roBots with Advanced Distributed Architec- ture
CERN	Conseil Européen pour la Recherche Nucléaire
CRIIS	Centre for Robotics in Industry and Intelligent Systems
DNMPC	Distributed Non-linear Model Predictive Controller
EKF	Extended Kalman Filter
ETL	ElectroTechnical Laboratory
EUA	Estados Unidos da América
FEUP	Faculdade de Engenharia da Universidade do Porto
FIFA	Federação Internacional de Futebol
FPS	Frames Per Second
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
INESC TEC	Instituto de Engenharia de Sistemas e Computadores - Tecnologia e Ciência
KF	Kalman Filter
MB	Megabyte
MSL	Middle Size League
NASA	National Aeronautics and Space Administration
ODE	Open Dynamics Engine
RoboCup	Robot World Cup Initiative
RTDB	Real-Time DataBase
SPAR	Strategic Positioning with Attraction and Repulsion
SSL	Small Size League
TU	Transformation of Uncertainty
TUE	Tech United Eindhoven
TURTLE	Tech United RoboCup Team: Limited Edition
UKF	Unscented Kalman Filter
UT	Unscented Transformation
ZMQ	ZeroMQ

Capítulo 1

Introdução

Neste capítulo será efetuada uma abordagem geral ao contexto do problema, assim como a motivação e objetivos desta dissertação.

1.1 Enquadramento

A robótica é uma área em crescimento que desperta interesse na comunidade científica, pela capacidade de apresentar soluções que realizam tarefas de forma mais eficiente e/ou que são potencialmente perigosas para o ser humano.

A cooperação entre robôs é um tema de elevada utilidade, podendo ser aplicada em diversas situações do dia-a-dia, tais como busca e salvamento, segurança e vigilância, mapeamento [1] e em contextos de investigação como o futebol robótico. A 25 de outubro de 2016, o departamento de defesa dos EUA realizou um conjunto de testes com um enxame de 104 drones, mostrando a sua capacidade de cooperação num vasto conjunto de missões [2]. Para que a interação entre os robôs ocorra da melhor forma, é necessário recorrer a um supervisor que gira as táticas e os papéis de cada interveniente e, que assegure uma correta fusão da informação obtida por cada agente. Da mesma forma, para que a cooperação seja realizada com sucesso, a correta deteção de alvos é crucial, pelo que a fusão sensorial toma também um papel bastante relevante nesta área.

1.2 Objetivos

A atribuição de táticas e papéis na cooperação de robôs não é uma tarefa trivial. Deve-se ter em atenção diversos fatores que podem variar no decorrer do processo, tal como falhas de comunicação, variações na sensorização e variação do número de agentes em cooperação. Assim, a presente dissertação tem como principal objetivo o desenvolvimento de um supervisor que coordene uma equipa de robôs, tendo em consideração estes fatores e que seja capaz de fundir a sensorização dos mesmos. Como caso de estudo e aplicação de teste com robôs reais, será utilizada a equipa de Futebol Robótico 5DPO da FEUP/INESC TEC.

1.3 Motivação

Uma equipa de robôs é composta por múltiplas máquinas que devem cooperar de forma a atingir um objetivo. No caso do futebol robótico, existem cinco robôs autónomos em campo, em que a equipa tem uma tática e cada robô tem o seu papel. Existe um programa, o treinador, que é executado num computador externo (*Base Station*), sendo responsável pela coordenação da equipa para que esta consiga cooperar dentro de campo. Da mesma forma, numa situação em que existe uma equipa de robôs de busca e salvamento ou de segurança e vigilância, é necessário haver coordenação entre os agentes envolvidos de forma a atingir um objetivo, seja ele a maximização da área de cobertura, minimização do tempo de execução da tarefa, etc.

Para coordenar uma equipa de uma forma eficiente é necessário fundir a informação de cada robô. Assim, consegue-se obter uma estimativa mais coerente e precisa da equipa e do ambiente que a rodeia em comparação com a estimativa que cada robô faria individualmente. No caso do futebol robótico, é necessário ter uma boa previsão da localização da bola, visto que esta nem sempre se encontra exatamente no local onde cada robô a percebe. Para além disso, é útil ter uma estimativa da velocidade e posição da mesma para melhorar a sua captação por parte do robô. Esta fusão também é realizada pelo supervisor. Isto também é aplicável em outras atividades do quotidiano, seja na localização mais precisa da pessoa que uma equipa de robôs está a socorrer, ou do intruso que estes pretendem identificar.

1.4 Caracterização do Problema

1.4.1 Definição do Problema

A crescente automatização de processos tem levado a um crescimento acelerado nas áreas da robótica e inteligência artificial. Inicialmente, era adotada uma arquitetura centralizada com um número limitado de agentes em uso. Contudo, o desenvolvimento tecnológico levou a que fosse possível ter mais robôs a operar em simultâneo, com capacidade de realizar tarefas mais complexas cumprindo requisitos temporais mais apertados. A cooperação entre agentes torna-se, assim, num requisito imprescindível, levando à necessidade da existência de um supervisor. Este deve ser robusto, ter a capacidade de fundir a informação obtida por cada agente, de adaptar a tática da equipa de acordo com as condições de operação em cada instante de tempo e de suportar um número variável de agentes em serviço. O problema em questão está presente nas diversas áreas de robótica, como a busca e salvamento, vigilância e mapeamento de ambientes. Embora haja ligeiras diferenças, a arquitetura principal e a filosofia de trabalho é partilhada pelas diversas aplicações.

Para além disso, a capacidade de um agente conseguir seguir um determinado objeto de interesse é cada vez mais um requisito nas mais diversas aplicações robóticas. Porém, o robô nem sempre está equipado com os sensores necessários para a medição das grandezas que se pretende medir, tornando-se, assim, necessário estimar essas grandezas a partir de outras cuja observação seja possível. Da mesma forma, as observações realizadas pelos agentes estão sujeitas a ruído que

pode ser prejudicial para a aplicação, ou o próprio objeto pode sair fora do campo de visão do agente que o está a seguir. Assim, para garantir uma boa qualidade de *tracking* é necessário recorrer a técnicas que permitam não só filtrar ruído, como também estimar as grandezas em análise no caso em que não existe observação (momentânea ou sistematicamente).

Esta dissertação tem como objetivo o desenvolvimento de um supervisor que contenha as características anteriormente descritas, sendo capaz de coordenar e fundir a informação de uma equipa de robôs de modo atingirem o objetivo pretendido. Deseja-se ainda que cada agente seja dotado da capacidade de obter uma boa estimativa de posição e velocidade de um objeto de interesse.

1.4.2 Solução Proposta

Como solução ao problema apresentado, propõe-se o desenvolvimento de um supervisor escrito numa linguagem de programação de alto nível, que contenha uma estrutura hierárquica de controlo, com clara distinção dos cinco níveis de planeamento e controlo (táticas, jogadas, papéis, tarefas e ações básicas). Como exemplo de aplicação, foi realizado um treinador de futebol robótico, devido à possibilidade de se realizarem testes em ambiente de simulação, assim como em ambiente real com os robôs da equipa de futebol MSL 5DPO. Embora o supervisor seja orientado para a sua aplicação (a nível de táticas, papéis e tarefas), este tem as características presentes num supervisor para qualquer outra aplicação, tais como a sua estrutura e capacidade de trabalhar com um número variável de agentes.

Para além disso, foram desenvolvidos dois mecanismos de fusão sensorial. O primeiro, implementado no treinador, teve como objetivo a estimação de grandezas como a posição e velocidade da bola tendo em conta a observação de cada jogador. Esta fusão é essencial, na medida que a posição da bola influencia a tática e jogada da equipa, assim como os papéis e tarefas atribuídos a cada agente. O segundo foi implementado no *software* de decisão que corre em cada agente. Este teve como objetivo a estimação de grandezas como a velocidade e posição da bola, tendo em conta apenas a observação feita por esse robô. Em ambos os casos foram utilizadas e comparadas algumas ferramentas de fusão robustas e computacionalmente leves como, por exemplo, o filtro de Kalman e algumas heurísticas.

1.5 Estrutura da Dissertação

No capítulo 2, é realizado o levantamento bibliográfico da temática desta dissertação. Inicialmente, são apresentados vários exemplos de cooperação em equipas de robôs móveis em diversas áreas de aplicação. Nas duas secções seguintes são abordados alguns exemplos e técnicas de fusão sensorial e, é apresentado o filtro de Kalman. A última secção aborda a maior competição de futebol robótico do mundo, assim como as diferentes estratégias de cooperação entre agentes, fusão sensorial e estimação de bola que várias equipas das diferentes ligas utilizam.

O capítulo 4 retrata a equipa de futebol robótico 5DPO, começando por fazer uma apresentação da mesma, seguindo-se duma breve explicação do *hardware*, sistema de visão e arquitetura de *software* e cooperação da mesma.

No capítulo 5, é apresentado o ambiente de simulação da equipa 5DPO, e são evidenciadas as alterações realizadas no mesmo.

No capítulo 6, é explicado o sistema de aquisição da medição da bola, e são apresentadas duas abordagens para a medida da posição e velocidade da bola por parte dos agentes da equipa de futebol robótico 5DPO. Primeiro é apresentada a implementação de um filtro de Kalman, em que o vetor de estado contém a posição e velocidades da bola no referencial local do robô. De seguida, é apresentado um segundo filtro de Kalman, desta vez com o vetor de estado referenciado no referencial global (do mundo). Para além disso, apresenta-se um terceiro método de estimação de velocidade da bola através duma regressão linear. Para as abordagens desenvolvidas, serão apresentados um conjunto de ensaios realizados em ambiente de simulação e ambiente real, através de dados recolhidos nos robôs da equipa 5DPO. Por fim, é apresentado o algoritmo de escolha de bola por parte do *software* de decisão, que corre em cada robô.

O capítulo 7 apresenta o novo treinador da equipa de futebol robótico 5DPO, desenvolvido na presente dissertação. Começará por se apresentar as funcionalidades e a arquitetura do mesmo, especificando-se, de seguida, a arquitetura do programa, a construção da sua representação do mundo, o ciclo principal de controlo, o mecanismo de atribuição dinâmico de papéis aos agentes e, finalmente, é justificada a adoção do novo sistema de comunicações da equipa.

No capítulo 8, são apresentadas duas abordagens para a fusão das bolas percecionadas pelos agentes, no treinador. A primeira abordagem consiste na utilização de um filtro de Kalman com o vetor de estado no referencial global, e serão testadas três abordagens diferentes para a escolha de bolas a utilizar na fase de correção do filtro. Para além disso, é apresentada uma segunda abordagem baseada numa heurística. Para ambas as abordagens, são apresentados um conjunto de testes realizados em ambiente real, com dados registados no treinador, através de observações realizadas pelos robôs da equipa de futebol robótico 5DPO.

Capítulo 2

Revisão Bibliográfica

Neste capítulo, será realizado o levantamento bibliográfico da temática desta dissertação. Começarão por ser abordados os tópicos de cooperação em equipas de robôs móveis e fusão sensorial, nos seus vários contextos de aplicação. De seguida, apresentar-se-á a iniciativa RoboCup, e serão abordados os temas anteriormente discutidos, aplicados no contexto do futebol robótico. Serão ainda referidas abordagens de equipas de diferentes ligas na temática de estimação de posição e velocidade da bola e obstáculos. Por fim, será apresentada uma ferramenta que se demonstrou ser bastante utilizada e útil no âmbito da fusão sensorial e *object tracking* - o filtro de Kalman.

2.1 Cooperação em Equipas de Robôs Móveis

O desenvolvimento científico e tecnológico a que se tem assistido nos últimos anos tem trazido novas soluções para diversos problemas do quotidiano. Atualmente, a robótica é uma área em grande crescimento, otimizando processos e sistemas, e tornando possíveis tarefas nunca antes exequíveis.

A cooperação de robôs móveis foi um dos fatores que contribuiu para o desenvolvimento desta área. Na verdade, uma equipa de robôs consegue cumprir uma tarefa em menos tempo do que se esta fosse realizada por um único robô. Além disso, a utilização de múltiplos agentes introduz redundância, tornando o sistema mais tolerante a falhas [3].

Burgard *et al.* [3] apresentam um exemplo de cooperação de robôs móveis na exploração de ambientes desconhecidos. A eficiência da exploração era maximizada através da variação da utilidade de exploração de áreas que já tinham robôs atribuídos e do custo de se chegar a essas áreas, reduzindo a probabilidade de múltiplos agentes explorarem a mesma área. Já Rocha *et al.* [4] propõem um modelo probabilístico para mapeamento 3D, com representação de incerteza. Esta abordagem revelou-se vantajosa na medida em que permitiu obter uma representação do mapa compacta, que era facilmente atualizada com novas medidas. A cooperação era realizada pela partilha de informação, permitindo construir o mapa em menos tempo do que com um único robô.

Outro domínio de aplicação é o da busca e salvamento [5], onde existe a necessidade de explorar a área onde ocorreu o desastre e, no caso de haver vítimas, de as transportar para um local seguro.

A cooperação de robôs em aplicações de vigilância é também um assunto de elevado interesse. Acevedo *et al.* [6] desenvolveram um sistema descentralizado, capaz de patrulhar uma determinada área em busca de intrusos, e de atribuir e distribuir a vigilância de intrusos pelos diversos agentes em serviço.

A nível industrial, têm sido desenvolvidos estudos com o intuito de ter um conjunto de robôs capaz de cooperar entre si, realizando tarefas mais complexas e de forma mais otimizada do que um conjunto de robôs a seguir um determinado escalonamento [7]. No artigo em questão, os agentes comunicavam através de uma rede *wireless*, a partir da qual trocavam informações de modo a fazerem um serviço de navegação (como mapeamento do local, localização cooperativa, planeamento de serviços e alocação de tarefas), e/ou operacional (logística de serviços e monitorização).

Matos e Cruz [8] apresentaram uma abordagem de cooperação entre veículos autónomos subaquáticos e de superfície, de forma a que estes se deslocassem em conjunto e, enquanto o primeiro estivesse a recolher dados, o segundo estaria a definir a posição do veículo subaquático. Também utilizando veículos subaquáticos e de superfície, Cruz *et al.* [9] desenvolveram uma rede acústica subaquática, baseando-se um algoritmo de cooperação centralizado, onde um líder virtual coordenava um conjunto de veículos e um ou mais seguidores executavam as ações instruídas pelo líder.

No âmbito do futebol robótico, Mota *et al.* [10] desenvolveram uma *framework*, denominada por *Setplay Framework*, que pudesse ser utilizada para planear o comportamento colaborativo entre agentes de uma equipa com esforço mínimo de programação.

2.1.1 Planeamento e controlo de tarefas

Para que haja uma boa cooperação entre uma equipa de robôs, é necessário que o sistema adote uma arquitetura de controlo robusta e bem definida. Os sistemas de controlo deliberativo seguem uma estrutura hierárquica, onde os níveis superiores definem os objetivos dos níveis inferiores. O planeamento e controlo de tarefas pode ser dividido em quatro níveis: táticas (*tactics*), papéis (*roles*), tarefas (*tasks*) e ações básicas (*actions*). O primeiro expressa o objetivo comum que a equipa pretende alcançar (realizar a tarefa no menor tempo possível, percorrer a menor distância, cobrir a maior área possível, etc). O supervisor tem uma noção geral do sistema e define os papéis a atribuir a cada robô. O segundo especifica o papel a realizar por cada robô (se está ativo, a carregar, de vigia, etc), decidindo as tarefas a atribuir ao mesmo. O terceiro nível define as tarefas que cada robô deve executar (furar uma peça, monitorizar uma determinada zona, ir para uma determinada posição). Por fim, o quarto nível especifica as atuações de baixo nível executadas pelo agente, e que permitem a realização de uma determinada tarefa (*GoToXYTheta*, *FollowLine*, *FollowParametricSegment*, *Stop*, etc) [11].

2.2 Fusão Sensorial e Seguimento de Alvos

A fusão sensorial é um tema de elevada importância em robótica. Um robô obtém uma percepção do espaço que o rodeia e da sua posição em relação ao mundo através de sensores. Contudo, estes estão sujeitos a ruído, apresentando uma dada incerteza nas suas medições. Torna-se, assim, imperativa a fusão de sensores para atenuar o efeito desse ruído na estimativa de um determinado estado. Vários estudos foram desenvolvidos nesta área, originando um conjunto de métodos capazes de resolver este problema, tais como filtros de Kalman, de média pesada, de partículas, entre outros.

Sasiadek e Hartana [12] desenvolveram um sistema de fusão de informação obtida através da odometria e sonares para estimar a posição de um robô, através de um filtro de Kalman Estendido (EKF) e um sistema de lógica difusa adaptativo (AFLS). A informação obtida pela odometria faria, em cada ciclo, a estimativa de estado. Quando o robô obtivesse uma medição de um sonar, o controlador *Fuzzy* iria adaptar os valores dos ruídos de medição e de estado, evitando, assim, que o filtro divergisse.

Outro exemplo de aplicação de fusão sensorial é a fusão de informação sensorial oriunda de vários agentes. Desta forma, é possível obter estimativas mais precisas, compensando a incerteza associada aos sensores. Karam *et al.* [13] desenvolveram um trabalho sobre a localização cooperativa de veículos heterogêneos. Cada veículo era capaz de se localizar a si em relação a um referencial absoluto, e aos veículos que o rodeavam em relação a si mesmo. Numa primeira fase, um veículo fazia uma atualização de estado com base na informação dos seus sensores. Numa segunda fase, eram medidas as posições relativas dos veículos mais próximos e obtidos os estados (posição e orientação) dos mesmos. Estes seriam transmitidos para os restantes veículos e fundidos com os mesmos.

Lima *et al.* [14] apresentaram uma solução que integrava um controlador e um estimador, de forma a obter-se melhores resultados durante uma percepção cooperativa de múltiplos robôs. Os autores sugeriram a utilização de um filtro de partículas capaz de estimar a posição e velocidade dos alvos, usando o resultado obtido para alimentar um controlador de formação que segue um modelo preditivo não linear (denominado por DNMP, *Distributed Non-linear Model Predictive Controller*). Este controlador era utilizado para fazer *tracking* de um alvo com base em percepção cooperativa, de tal forma que minimizasse a incerteza da observação da posição do alvo, mantendo uma distância de observação desejada e evitando colisões entre os agentes presentes na formação.

Ferrein *et al.* [15] apresentaram uma comparação de diferentes métodos de fusão sensorial, aplicados à estimação de posição da bola por parte de uma equipa de robôs. Verificou-se que os métodos que apresentaram melhores resultados consistiam em dois filtros de Kalman (um convencional e outro em que reiniciavam a matriz de covariância do erro P de cada vez que a posição da bola fosse calculada), e um método de Monte Carlo. Os autores destacaram ainda a rapidez e exatidão da posição estimada pelo filtro de Kalman, assim como a velocidade de relocalização de uma bola desconhecida por parte do filtro de Kalman com *reset*.

2.3 Futebol Robótico

2.3.1 Robocup

O futebol robótico nasceu na década de 90. Em 1992, o Professor Alan Mackworth, da *University of British Columbia* no Canadá, publicou um artigo que idealizava robôs a jogar futebol. Em outubro do mesmo ano, um grupo de investigadores japoneses organizou um *workshop* onde se discutiu a utilização do futebol para promover a ciência e tecnologia. Um ano mais tarde, Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi e outros lançaram a competição robótica *Robot J-League*. Num curto espaço de tempo, houve pedidos no sentido de tornar esta iniciativa internacional, resultando na renomeação do projeto para *RoboCup (Robot World Cup Initiative)*. Paralelamente, realizava-se uma investigação sobre o uso de multi-agentes no futebol no laboratório japonês *ElectroTechnical Laboratory (ETL)* e de experiências com robôs a jogar futebol na *Carnegie Mellon University* e no laboratório Minoru Asada, na *Osaka University* [16].

A Robocup é uma iniciativa que pretende promover a investigação em robótica e inteligência artificial, tendo como principal objetivo a criação de uma equipa de robôs futebolistas totalmente autónoma que, a meio do século XXI, consiga ganhar à equipa (humana) campeã do mundo [17]. Neste momento, esta iniciativa conta com várias ligas, nomeadamente *RoboCup Industrial*, *RoboCup Junior*, *RoboCup@Home*, *RoboCup Rescue* e *RoboCup Soccer*. Esta última é composta pelas ligas *Humanoid*, *Standart Plataform*, *Middle Size*, *Small Size* e *Simulation* [18, 19].

Middle Size League

Os jogos da liga MSL são disputados por duas equipas, com no máximo cinco agentes completamente autónomos em campo. Os robôs não devem exceder as dimensões $52 \times 52 \times 80$ cm (exceto o guarda-redes que, no caso de uma bola se estar a aproximar, pode exceder-se até $60 \times 60 \times 90$ cm, durante o máximo de 1 segundo e nunca num tempo inferior a 4 segundos após o último recolher ao tamanho original). Devem ter um peso inferior a 40kg, e estar equipados com os sensores necessários para se localizarem e obterem uma perceção do estado do mundo à sua volta. A sua cor base é o preto, e devem estar numerados com marcadores de qualquer cor, exceto branco, preto, verde e a cor da bola, visíveis em todas as suas faces. A bola de jogo deve ter tamanho 5, segundo o *standard* da FIFA, e pode ser de qualquer cor exceto branco, preto ou verde (tipicamente são utilizadas bolas amarelas ou laranja). O campo tem dimensões 18×12 m, é de cor verde, com linhas brancas e uma barreira de segurança preta, que evita que os robôs saiam de dentro do mesmo.

Cada partida é composta por duas partes de 15 minutos, e são arbitradas por humanos, segundo adaptações das regras da FIFA ¹. Para que as decisões tomadas sejam compreendidas pelos agentes e, a fim de manter dados estatísticos sobre o jogo e equipas, é utilizado um programa auxiliar conhecido por *Referee Box*.

A nível de *software*, as equipas são livres de escolher a sua arquitetura, devendo ter um computador externo, denominado por *Base Station*, que deverá fazer a interligação entre a *Referee Box*

¹http://wiki.robocup.org/images/7/72/2017-msl-rules_v18_4.pdf

e os seus agentes. As comunicações são realizadas via *wireless* (exceto entre as *Base Stations* e a *Referee Box*), devendo passar sempre obrigatoriamente por um dos *Access Points* existentes, e respeitar os limites de tráfego estabelecidos (20% da largura de banda disponível). Apenas é permitida interação humana de alto nível, através de gestos ou voz. De outra forma, apenas é permitido mexer nos robôs quando os mesmos se encontram fora de campo, desativados [20].

Esta liga aborda desafios das mais vastas áreas, desde localização, visão, percepção, fusão sensorial, cooperação multi-agente, entre outros. É, decerto, um desafio complexo e interessante, que várias equipas a nível mundial tentam resolver.

2.3.2 Estratégias de Cooperação Entre Agentes

O futebol é um desporto de equipa, em que todos os jogadores têm de cooperar por forma a realçar as suas capacidades e explorar os espaços da equipa adversária. Consequentemente, a capacidade de uma equipa de robôs conseguir cooperar é uma característica crucial e decisiva durante um jogo. Vários estudos têm sido feitos no sentido de melhorar a coordenação multi-agente, assim como o posicionamento e distribuição de tarefas/papéis durante um jogo.

Uma das primeiras abordagens de coordenação de alto nível no contexto do futebol robótico SSL foi apresentada em 1998 por Veloso *et al.* [21]. O algoritmo utilizado pelos CMDragons antecipava oportunidades de colaboração, aumentando, assim, as oportunidades de passe. Cada robô poderia ter um de dois estados: ativo ou passivo. Enquanto um robô ativo tinha uma tarefa específica para executar, como ir para a bola, os robôs passivos antecipavam oportunidades de colaboração, posicionando-se em posições estratégicas que aumentassem as oportunidades de passe. Estas posições eram calculadas com base num algoritmo semelhante a um campo de potencial, denominado por SPAR (*Strategic Positioning with Attraction and Repulsion*), que maximiza a repulsão dos adversários e minimiza a atração para a bola e a baliza adversária. Eram ainda adicionadas restrições para que não fosse calculada uma posição que fosse atrás de adversários (com o intuito de receber um passe), e que não bloqueasse o remate de um companheiro de equipa. Em 2015, Mendonza *et al.* [22] apresentaram um algoritmo de coordenação baseado por zonas, com posicionamento reativo. A equipa era dividida numa subequipa de N robôs defensivos, e numa subequipa de R robôs atacantes. A primeira, tinha como objetivo minimizar a probabilidade da equipa sofrer um golo, enquanto que a segunda pretendia maximizar a probabilidade de marcar. O algoritmo propõe ainda a criação de $R - 1$ zonas. Um dos robôs ofensivos teria o papel *Primary Attacker* e estaria encarregue de manter a posse de bola, rematar à baliza adversária ou passar a um companheiro de equipa. Os restantes $R - 1$ robôs ficariam com o papel *Support Attacker*, e seria atribuída uma zona Z_i a cada. Se o *Primary Attacker* estivesse pronto para realizar um passe a um agente *Support Attacker*, este tentaria encontrar uma posição ótima para receber o passe, dentro da zona Z_i , senão iriam para uma das posições de guarda, previamente definidas na zona. Por sua vez, o algoritmo responsável pela coordenação defensiva baseava-se num detetor de ameaças, capaz de caracterizar um robô adversário na iminência de remate como ameaça do tipo 1, e robôs capazes de receber um passe e rematar diretamente como ameaças do tipo 2. Dos N robôs defensivos, um ou dois seriam o(s) *Primary Defenders* e deslocar-se-iam para junto da área defensiva. Já os

restantes, teriam o papel *Secondary Defenders*, estariam mais avançados no campo e tentariam interceptar os passes e remates adversários. Browning *et al.* [23] desenvolveram uma arquitetura hierárquica para uma equipa de robôs móveis em ambientes adversos. Esta arquitetura tinha por base três conceitos chave: *skills*, *tactics* e *plays*. Desta forma, a equipa tinha um plano com um conjunto de papéis a desenrolar pelos robôs (*play*), onde cada papel tinha um conjunto (*tactics*) de habilidades (*skills*) a executar. Como a finalidade desta investigação consistia na aplicação desta arquitetura num ambiente dinâmico, o algoritmo responsável pela execução das jogadas tinha a capacidade de ajustar a distribuição dos papéis pelos robôs ativos.

No âmbito da *Simulation League*, Reis *et al.* [24] apresentaram duas abordagens de posicionamento de agentes. Na primeira, *Situation Based Strategic Positioning*, os agentes que se encontravam numa situação ativa decidiam as suas ações com base num mecanismo de decisão reativo, enquanto que os restantes tentavam ocupar posições estratégicas que maximizassem as opções de cooperação com agentes em situação ativa e mantivessem a equipa distribuída pelo terreno de jogo. Esta posição era calculada tendo em conta a situação do jogo e a posição estratégica base do agente para uma determinada formação, sendo ainda ajustada de acordo com a posição e velocidade da bola [24, 25]. A segunda abordagem denomina-se por *Dynamic Positioning and Role Exchange*, e consistia na troca de posição e papel de dois agentes, numa determinada formação. Este método baseava-se no facto de cada agente conhecer não só a sua composição, como também a dos restantes, avaliando constantemente se uma troca de posição seria vantajosa para a equipa. Essa avaliação tinha como critérios a distância da posição atual do robô para a sua posição estratégica, assim como a importância do seu posicionamento na formação em uso [24, 25]. Em 2002, a equipa portuguesa apresentou uma linguagem, *Coach Unilang*, que permitia "treinar" uma equipa de futebol robótico. Esta linguagem permitia definir a tática, formações e o comportamento dos jogadores para cada formação. Para além disso, possibilitava o envio de informação sobre os adversários e de estatísticas do jogo para os agentes da equipa [26].

Em 2003, Spaan e Groen [27] apresentaram a noção de tática de equipa e o sistema de atribuição dinâmica de roles utilizado pela equipa Clockwork Orange. A estratégia de jogo era decidida de acordo com o estado da bola, e definia um conjunto de papéis a serem distribuídos pelos agentes ativos. Para a atribuição dos papéis, a equipa recorria a *utility functions* baseadas no tempo que o agente demora a atingir a bola, e na adequabilidade da posição do robô ao papel em questão. Adicionalmente, a equipa decidiu atribuir uma prioridade a cada papel de modo a garantir que os papéis mais importantes continuavam a ser atribuídos no caso de haver uma redução no número de robôs ativos em campo.

No que concerne à coordenação de alto nível, a equipa CAMBADA [28] apresentou uma arquitetura de *software* que toma decisões com base em informação presente e passada, e prevê ainda estados futuros. O algoritmo de atribuição de papéis corre localmente em cada robô, e tem em atenção um conjunto de escolhas com certas condições e prioridades [28]. Em 2011, Lau *et al.* [29] apresentaram um sistema de atribuição dinâmica de roles. Cada *role* era modelizado através duma máquina de estados finita, e definia quais os comportamentos (*behaviors*) do robô em cada instante, com base no estado atual e no estado do mundo. A atribuição de *roles* era calculada

localmente em cada robô, e tinha por base diferentes prioridades para diferentes roles e posições, tentando que os *roles* mais importantes fossem cobertos. Durante uma jogada normal, o jogador mais perto da bola ficava com o papel *RoleStriker*, enquanto que os restantes ficavam com o papel *RoleSupporter*. O guarda-redes desempenhava sempre o mesmo papel (*RoleGoalie*). No que concerne ao posicionamento, eram calculadas as distâncias a pontos estratégicos, e o robô com menor distância assumia o lugar correspondente na formação. Nesse ano, a equipa apresentou dois papéis novos que permitissem a execução de passes: *RolePasser* e *RoleReceiver*. Desta forma, sempre que se quisesse realizar um passe, os agentes com esses papéis trocavam *flags* que ajudariam na sincronização. Em situações de *set pieces*, a equipa introduziu os papéis *RoleBarrier*, *RoleReplacer* e *RoleToucher*. Como o *RoleReplacer* deve ser um papel único, durante o processo de seleção desse *role* os agentes verificavam se não existe nenhum robô com um número inferior que já tinha esse papel atribuído. Nesse caso, esse *role* não era escolhido. Os agentes voltavam aos papéis existentes na jogada normal assim que passasse um *timeout*, ou que a ação da *set piece* tivesse sido executada com sucesso [29]. Em 2015, Neves *et al.* [30] desenvolveram um algoritmo de posicionamento estratégico baseado em *Utility Maps*. Os autores começam por sugerir a criação de um mapa de ocupação com o estado do mundo à volta do agente. De seguida, com base na situação do jogo e papel ativo, era calculado o campo de visão (*Field of Vision*) sobre o mapa de ocupação. Por fim, era criado o *Utility Map* com base no mapa de ocupação, campo de visão e certas condições que dependiam da situação de jogo. Desta forma, os robôs determinavam qual a melhor posição para ocupar em campo através do cálculo dos máximos locais do mapa. A equipa apresentou ainda o seu treinador como sendo a aplicação que mantinha o sincronismo de posições estratégicas e definia qual a melhor formação em cada momento. Embora não haja interação com os agentes durante um jogo, a equipa conseguia configurar as formações disponíveis, assim como definir um conjunto de regras associado a cada formação, através de uma aplicação denominada *CoachConfig* [31]. A interligação dos agentes da equipa com a *Referee Box* está a cargo dum *software* denominado *Base Station*. Esta permite monitorizar o estado do jogo e o estado interno do robô, nomeadamente a posição em campo, velocidade e bateria. Os autores afirmam ser também uma ferramenta de *debug*, permitindo enviar comandos básicos aos agentes (por exemplo *Run*, *Stop*), forçar jogadas, papéis, entre outros [32]. Além disso, a aplicação tem um mecanismo de *replay* que, a qualquer momento, permite parar os agentes e analisar as situações anteriores. A *Base Station* permite ainda guardar *logs* durante o jogo e, mais tarde, carregar um vídeo e sincronizar a sua visualização com os *logs* anteriormente guardados [33].

Até 2017, os Tech United Eindhoven utilizavam uma estratégia ofensiva, que simplesmente atribuía papéis aos agentes ativos. Durante o período de jogo, contavam com um guarda-redes, dois defesas e dois atacantes, sendo que cada papel evocava um conjunto de tarefas, que por sua vez selecionava e executava habilidades necessárias à execução das mesmas. Contudo, esta estratégia não se adaptava ao estado do jogo, e permitia que os adversários que defendiam com um elevado número de robôs dificultassem a concretização das jogadas [34]. Assim, a equipa holandesa implementou nos seus *TURTLEs* a arquitetura de *skills*, *tactics* e *plays* desenvolvida pelos CMDragons [35]. Nesta arquitetura, o plano da equipa e papéis desempenhados por cada

agente eram definidos pelas jogadas. As táticas determinavam e geriam um conjunto de tarefas que cada jogador deveria executar individualmente em prol do objetivo atual da equipa. As habilidades estabeleciam o comportamento que cada agente desempenha, tal como ir de encontro à bola, driblar, etc. [35] As jogadas eram selecionadas com base no estado do jogo, tal como o número de obstáculos e posição da bola no campo. Porém, devido ao facto da equipa holandesa utilizar uma arquitetura distribuída, onde cada robô tem uma percepção diferente do mundo, era necessário garantir que todos os agentes chegam a um consenso acerca da jogada e distribuição de papéis a utilizar. Para isso, Koning *et al.* [34] recorreram a um sistema de *voting*, onde a atribuição de papéis era realizada dinamicamente com base numa função custo baseada em distâncias. Contudo, o facto da equipa Tech United usar uma arquitetura distribuída, podia levar a que os agentes tivessem diferentes percepções do mundo, levando diferentes matrizes de custo e, consecutivamente, uma atribuição de papéis diferente. Para contornar este problema, a equipa propôs duas soluções: uma que garantia menor complexidade (a nível de transmissão de dados), e outra que garantia a solução ótima. Na primeira, todos os agentes calculavam a atribuição de roles ótima e comunicavam em *broadcast* para toda a equipa. Contudo, todos os robôs assumiam a atribuição oriunda do primeiro robô ativo. Na segunda, cada agente apenas calculava e partilhava a sua matriz de custo. De seguida, procedia-se à fusão destas matrizes numa única, sendo esta utilizada para o cálculo local da atribuição dos *roles*, em cada agente. Adicionalmente, a equipa apresentou também uma nova abordagem defensiva que definia um índice de perigo que cada robô adversário apresentava, associando-o a um conjunto de propriedades geométricas (posse de bola, distância e ângulo em relação à baliza, oportunidade de passe, etc). Desta forma, a defesa a cada adversário era adaptada a quão perigoso aparentava ser [35].

2.3.3 Estratégias de Fusão Sensorial

Um dos principais requisitos para obter um bom desempenho num jogo de futebol robótico é a capacidade de obter um bom conhecimento do estado do mundo. Uma equipa pode ser constituída por agentes homogêneos ou não, podendo, portanto, ter capacidades de sensoramento diferentes. Da mesma forma, dois agentes em posições distintas podem fazer diferentes observações, ou até nem conseguir realizar nenhuma. Por estes motivos, têm sido desenvolvidos vários estudos na temáticas da fusão sensorial, alguns deles no âmbito do futebol robótico [33, 36–39].

Recorrendo a um filtro de Kalman, Stroupe *et al.* [36] demonstraram que se conseguem obter uma melhor estimativa da posição de um objeto quando se recorre à fusão sensorial de múltiplos robôs do que à observação de um único.

Ahmad *et al.* [40] apresentaram um estimador cooperativo baseado que utilizava um filtro de partículas para fundir a observação de um alvo, fornecendo a posição e velocidade do mesmo. Os autores propuseram ainda um controlador preditivo não linear para controlar a formação de equipa de robôs, de forma a minimizar a incerteza da percepção de um alvo.

A equipa CAMBADA utiliza uma base de dados de tempo real (RTDB) para partilhar informação relevante entre os agentes [33]. Para permitir um sensoramento cooperativo, Almeida *et*

al. [41] desenvolveram uma estrutura de dados partilhada, implementada sobre um bloco partilhado de memória. Este bloco contém uma área privada, somente para informação local, e uma área partilhada dividida em pequenas áreas que contém a informação de cada agente da equipa. Uma dessas áreas é escrita pelo agente local, enquanto as restantes são usadas para armazenar a informação oriunda dos restantes agentes. Este era um conceito semelhante a um *blackboard* (base de dados onde vários agentes publicam a sua informação, que pode ser requisitada por outros agentes). Porém, na RTDB o estado local é disseminado em *broadcast*, usando o modelo de cooperação produtor-consumidor. Para além disso, o "*blackboard*" é replicada de acordo com o modelo de memória partilhada distribuída dando a cada nó acesso local às variáveis de estado do processo. Existe ainda um sistema de comunicação, que garante que as imagens são atualizadas e evita a perda de pacotes de rede devido a colisões.

Em 2010, a estrutura da bola continha o número de ciclos em que a bola não era visualizada por parte do robô. Quando a bola não era observada por mais do que um determinado número de ciclos, o agente assumia que não conseguia visualizar a bola, passando a usar a bola partilhada pelos companheiros de equipa, nomeadamente a bola do agente que tem mais confiança na observação [37]. Para além da bola, a CAMBADA é capaz de identificar obstáculos que, durante um jogo, são robôs adversários [38]. As regras desta liga impõe que todos os robôs tenham como cor base o preto [20], tornando assim necessário um tratamento extra dos possíveis obstáculos que um robô deteta, de modo a serem identificados como obstáculos ou companheiros de equipa. Para contornar este problema, a equipa da Universidade de Aveiro propõe uma fusão entre a observação dos candidatos a obstáculos com as posições dos agentes, partilhada entre os mesmos via RTDB. Assim, criaram um círculo em torno da posição de cada companheiro de equipa e, no caso do de haver uma interceção dos pontos dos obstáculos com esse círculo (ou se o obstáculo estiver contido dentro desse círculo), esse candidato era identificado como um agente da própria equipa. Caso contrário, era identificado como um obstáculo. Para que um *blob* fosse detetado como obstáculo, tinha de estar dentro de campo e cumprir certos limites de tamanho. Se porventura um candidato a obstáculo excedesse as dimensões máximas, era feita uma análise do número de robôs que podiam estar dentro desse candidato, dividindo-o [38].

Em 2017, Dias *et al.* [39] apresentaram outra abordagem. A equipa introduziu o conceito de *tracklets* como sendo objetos a seguir, nomeadamente a bola e obstáculos. Depois de processados², os *tracklets* eram partilhados com a restante equipa. Como existem limites na largura de banda que cada equipa pode utilizar nas comunicações, procedia-se a uma ordenação dos *tracklets* de acordo com os seguintes critérios: no caso dos obstáculos, eram ordenados por ordem decrescente de visibilidade, dando prioridade de partilha aos *tracklets* com maior visibilidade; no caso das bolas, eram ordenadas por ordem decrescente de idade, sendo partilhadas apenas as bolas com maior idade. Para além disso, para garantir a precisão da informação, apenas seriam partilhados *tracklets* com uma idade, número de observações e taxa de visibilidade superiores a determinados *thresholds*, e que não estavam além de uma determinada distância do robô que realizou a observação.

²O processamento dos *tracklets* será abordado mais pormenorizadamente na secção 2.3.4

2.3.4 Estratégias de Estimação de Posição e Velocidade da Bola

Tal como num jogo de futebol convencional, grande parte das decisões tomadas durante um jogo de futebol robótico têm por base no estado da bola. Esta influência pode ir desde a escolha da tática da equipa, atribuição de papéis, ou até mesmo ao comportamento dos agentes. Este é um desporto onde é preciso ir de encontro à bola, fazer/receber passes, driblar e rematar à baliza. Para que isto seja possível, é preciso conhecer a posição e velocidade da bola em cada instante de tempo, com a melhor precisão possível. A resolução deste problema não é trivial, principalmente na liga MSL, visto que a grande maioria dos robôs apenas está dotado de um sistema de visão catadióptrico, que apenas lhes permite fazer uma observação da posição da bola. Contudo, esta nem sempre está no campo de visão dos robôs, e a própria observação está sujeita a ruído que não pode ser desprezado. Por estes motivos, tem havido um grande esforço por parte das equipas das diferentes ligas de futebol robótico por forma a melhorar a observação da bola, e conseguir obter uma estimativa da sua velocidade. São várias as abordagens das equipas das diferentes ligas, contudo, é notória uma preferência pela utilização do filtro de Kalman [21, 37, 39, 42–45].

Em 2010, a equipa CAMBADA apresentou uma abordagem que consistia na utilização de um filtro de Kalman para obter a estimativa da posição da bola, e de uma regressão linear para a estimativa da velocidade da mesma. Partindo do princípio que a velocidade da bola se mantém constante, que não existe nenhum controlo sobre a mesma e, que esta não sofre ação do atrito, a equipa desenvolveu um filtro de Kalman com o modelo de transição de estado presente na equação (2.1). O vetor de estado continha a posição e velocidade da bola nas componentes x e y , sendo que estas velocidade seriam apenas internamente estimadas pelo filtro. Para modelizar o ruído de medição a equipa aproximou o desvio padrão da medição da posição da bola em função da distância por um polinómio de segundo grau. Já o ruído do modelo de transição de estado foi ajustado de modo a se obter um compromisso entre suavidade e reatividade. Como optaram pela utilização de um modelo de movimento uniforme, a velocidade estimada pelo filtro não era muito precisa [43]. Por outro lado, o cálculo da velocidade utilizando a derivada de primeira ordem da posição, por intervalo de tempo, está muito sujeito a ruído, levando a resultados indesejáveis. Para contornar este efeito, a equipa decidiu guardar as observações da bola e o respetivo instante de tempo num *buffer* e, fazendo uma regressão linear com esses pontos conseguiu obter uma estimativa de velocidade mais precisa e com menos ruído. A partir diferença entre a posição da bola estimada pelo filtro de Kalman da realmente medida era possível detetar desvios acentuados. Quando estes desvios eram detetados, o tamanho do *buffer* era reduzido, fazendo com que a regressão linear convergisse mais rapidamente [37].

$$X(k) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} X(k-1) \quad (2.1)$$

Em 2017, a mesma equipa apresentou um algoritmo de *tracking* de objetos (mais concretamente, bola e obstáculos) [39]. Este algoritmo modelizava um processo gaussiano num plano 2D, e utilizava um filtro de Kalman para estimar as posições e velocidades dos objetos segundo

as componentes x e y (x, y, \dot{x}, \dot{y}). Os objetos a seguir foram denominados de *tracklets* e, adicionalmente ao vetor de estado anteriormente referido, eram guardadas informações acerca da idade e número de ciclos consecutivos de observação e não observação (*age*, *visibleCount* e *invisibleCount*). O algoritmo *object tracker* era atualizado a cada ciclo, e podia ser dividido em sete fases. Na primeira era corrida a fase de predição do filtro de Kalman de cada *tracklet*, atualizando-se as matrizes de covariância e a estimativa de estado. Na segunda, era feita uma atribuição das observações aos *tracklets*. Na verdade, nem todas as observações correspondem a um *tracklet* já existente, podendo assim ser criados novos *tracklets*, ou até mesmo existir *tracklets* sem nenhuma observação. Para fazer estas correspondências, os autores propuseram a utilização do método Húngaro que, assumindo que apenas existe uma observação por objeto, encontrava o valor ótimo para uma matriz de custo C . Este custo consiste numa matriz quadrada cujo tamanho se ajusta ao número de *tracklets* (T) e observações (N), e em que as linhas correspondem ao *tracklets* e as colunas às observações. Depois do algoritmo ter corrido resultava uma matriz C' , que continha um grupo de observações que correspondiam a um *tracklet* (M), que não correspondiam a nenhum *tracklet* (V) e *tracklets* que não correspondiam a nenhuma observação (U) (2.2). Na terceira fase, os filtros de Kalman de cada *tracklet* eram atualizados com a respetiva observação, e eram atualizados os campos *age*, *visibleCount* e *invisibleCount*. Na quarta, eram atualizados os *tracklets* sem observação, e multiplicando a velocidade por um fator inferior a um, para que esta não se mantivesse constante neste intervalo de tempo. Na quinta fase, eram apagados os *tracklets* que tivessem um *invisibleCount* maior que um determinado valor, ou uma taxa de visibilidade (*visibleCount/age*) inferior a dado *threshold*. Na sexta fase, eram criados novos *tracklets* que resultassem de observações sem nenhum *tracklet* associado (submatriz V) e, finalmente, na sétima fase, os *tracklets* eram ordenados de acordo com um determinado critério para que apenas os *tracklets* mais importantes fossem partilhados com a equipa. No caso dos obstáculos, eram ordenados por ordem decrescente de visibilidade, de modo a serem apenas partilhados os obstáculos com maior taxa de visibilidade. No que concerne à bola, como um agente pode observar mais do que uma bola, estas eram ordenadas por ordem decrescente de idade, sendo que o agente assumia que a bola certa era a bola mais antiga.

$$C' = \begin{bmatrix} M & U \\ V & 0 \end{bmatrix} \quad (2.2)$$

Existem ainda abordagens que utilizam técnicas diferentes. Ahmad e Lima [46] desenvolveram uma perceção cooperativa que aplica um algoritmo de localização da bola num plano 2D, baseado num filtro de partículas. Para isso, calculavam-se os fatores de confiança associados ao objeto a detetar tendo em conta a confiança na observação e da estimativa de localização de cada robô. O modelo de observação de cada sensor era uma função densidade de probabilidade. As funções de probabilidade associadas a cada robô eram partilhadas por todos os agentes numa *pool* e cada um seleccionava a melhor função (onde os fatores apresentavam maior confiança) e utilizavam-na para o cálculo do peso de cada partícula. Como a função densidade de probabilidade podia ser identificada pelos parâmetros comunicados, conseguia-se reduzir significativamente a

quantidade de informação trocada por cada agente. No mesmo contexto de aplicação, Lauer *et al.* [47] desenvolveram um estimador para a posição e velocidade da bola que apenas assumia que as observações eram realizadas com ruídos da mesma ordem de grandeza, não necessitando de ter uma estimativa do mesmo. Considerando um movimento livre da bola, a posição e velocidade seriam estimadas simplesmente com base numa *ridge regression*. Porém, o modelo considerado deixava de ser válido no caso de haver uma mudança brusca no movimento da bola (por exemplo, devido a uma colisão). Para cobrir esse caso, foi adicionado um mecanismo de detecção de mudanças bruscas a partir da comparação das posições observadas e estimadas da bola. Nesse caso, eram descartados os pontos anteriores e o modelo de movimento seria novamente calculado com base nas novas observações.

Até agora foram apresentadas abordagens que supunham um modelo da bola apenas no plano horizontal (a duas dimensões). Todavia, existem inúmeras ocasiões em que o esférico se encontra acima do nível do solo, sendo de total interesse o conhecimento da sua altura, a fim de conhecer a sua trajetória e ponto de embate com o chão. A equipa CAMBADA, apresentou duas soluções para a estimação da posição da bola num plano 3D para os seus robôs da liga MSL [28]. A primeira era aplicada ao guarda-redes que, para além do sistema de localização principal (composto com uma câmara com visão omnidireccional), continha um sensor Kinect, da Microsoft®. A partir da profundidade obtida por este sensor conseguia-se obter a posição da bola num plano 3D, possibilitando um melhor posicionamento para as defesas. Para os restantes robôs, a equipa desenvolveu uma solução mais inovadora, em conjunto com a equipa holandesa Tech United Eindhoven [48]. Cada agente estava equipado com um sistema de visão catadióptrico (omnidireccional), e partilhava a posição da bola com os restantes robô, obtendo a posição no plano tri-dimensional da bola através da triangulação da informação obtido pelas diversas câmaras. Adicionalmente, um filtro de Kalman, que fundia as medidas com um modelo de bola híbrido. A informação adquirida pelo sistema de visão era fundida com a informação dos restantes companheiros de equipa, nomeadamente a sua posição no campo, assim como a posição da bola e posição e classificação de obstáculos (companheiros de equipa ou adversários) [30].

Este problema é igualmente desafiante na *Small Size League*. Nesta liga, o sistema de visão é ligeiramente diferente, existindo duas câmaras localizadas a quatro metros de altura acima da superfície de jogo, fornecendo uma visualização de todo o campo. O jogo desenrola-se com uma bola de golf laranja sobre um campo verde [49]. Embora a observação da bola seja mais acessível, existe na mesma a necessidade de manter *tracking* da posição e velocidade da mesma.

Em 1999, Costa [45] sugeriu a implementação de um filtro de Kalman para estimar a posição e velocidade da bola nos robôs da geração SSL da equipa 5DPO. Devido aos baixos valores típicos do coeficiente de atrito de uma bola de golf, o autor desprezou este efeito, adotando um modelo

de velocidade linear com o seguinte modelo estocástico:

$$\begin{aligned}
 \dot{x}_b &= v_{bx} + \omega_{bpx} \\
 \dot{y}_b &= v_{by} + \omega_{bpy} \\
 \dot{v}_{xb} &= -\lambda_b v_{bx} - \nabla_x h(x, y) + \omega_{bv_x} \\
 \dot{v}_{yb} &= -\lambda_b v_{by} - \nabla_y h(x, y) + \omega_{bv_y}
 \end{aligned} \tag{2.3}$$

onde v_{bx} (\dot{x}_b), v_{by} (\dot{y}_b), v_{xb} e v_{yb} são respetivamente as velocidades e acelerações da bola nas componentes x e y , ω_{bpx} , ω_{bpy} , ω_{bv_x} e ω_{bv_y} são o ruído do processo da posição e velocidade, e $h(x, y)$ é o desvio da altura do campo em relação à horizontal para as coordenadas x e y .

Para a covariância do ruído de processo da velocidade $\omega_{bv[x,y]}$ foi utilizado um valor baixo. Já para a covariância de processo da posição $\omega_{bp[x,y]}$, o autor assumiu que esta variava ao longo do tempo, tomando um valor baixo no caso da bola se deslocar livremente (isto é, longe das paredes e dos restantes robôs), e elevado nos instantes em que é iminente uma mudança de direção da bola. Desta forma, nestes instantes era possível dar mais importância às observações do que à dinâmica do modelo, obtendo resultados mais precisos.

Assim, foi possível obter filtragem da posição e um estimador de estado que fornecia a velocidade segundo as componentes x e y . Este último fornecia um elemento preditivo ao sistema global de controlo, permitindo prever a posição da bola em instantes de tempo futuros [45].

2.4 Filtro de Kalman

Na vida real, um sistema está sujeito a perturbações que, normalmente, são sinais desconhecidos. O Filtro de Kalman (KF) é um filtro linear discreto e, eventualmente, variante no tempo que, através de ciclos consecutivos de predição e filtragem, estima o vetor de estado do sistema e minimiza o erro médio quadrático dessa estimativa. Estas perturbações podem ser adicionadas ao modelo do sistema como dois fatores aditivos, resultando no modelo:

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) \tag{2.4}$$

$$y(k) = C(k)x(k) + v(k) \tag{2.5}$$

onde $x(k) \in \mathbb{R}^n$ e $x(k+1) \in \mathbb{R}^n$ são vetores de estado no instante k e $k+1$, $u_k \in \mathbb{R}^m$ é o vetor com os sinal de entrada, $y(k) \in \mathbb{R}^r$ é o vetor dos sinais de saída, $w(k) \in \mathbb{R}^n$ é o ruído de estado, $v(k) \in \mathbb{R}^r$ é o ruído de medição, $A(k) \in \mathbb{R}^{n \times n}$ é a matriz de transição estado, $B(k) \in \mathbb{R}^{n \times m}$ é a matriz de entrada e $C(k) \in \mathbb{R}^{r \times n}$ é a matriz de saída. A figura 2.1 apresenta o diagrama de operação do filtro de Kalman.

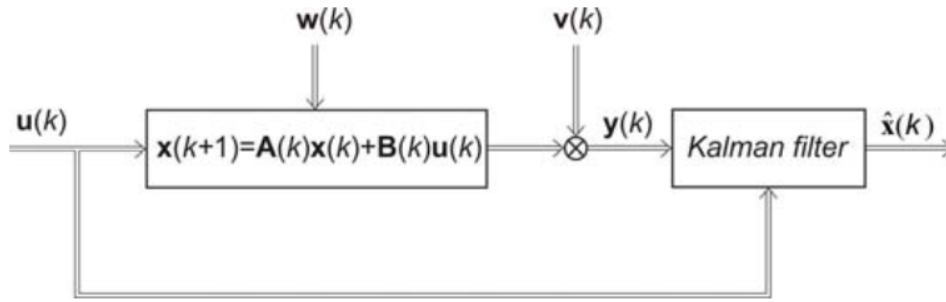


Figura 2.1: Diagrama de operação do filtro de Kalman [50]

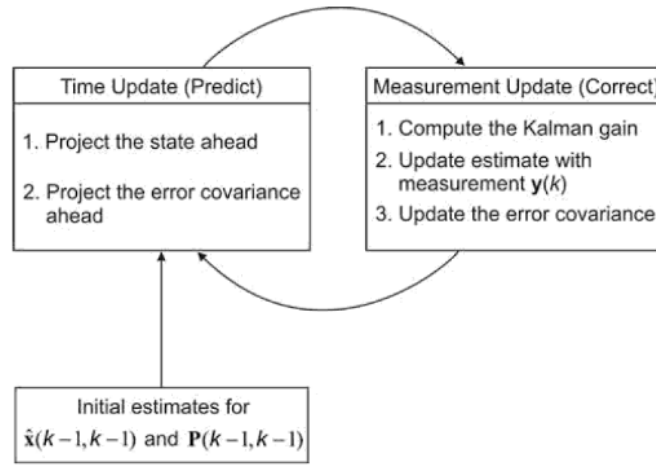


Figura 2.2: Algoritmo de operação do filtro de Kalman [50]

Na primeira iteração, é necessário introduzir no filtro os vetores estado (x_0) e covariância (P_0) iniciais. O filtro começa por estimar o estado seguinte (equação (2.6)) e a covariância do erro dessa estimativa (equação (2.7)) (estado de predição)).

$$\hat{x}(k+1|k) = A(k)\hat{x}(k|k) + B(k)u(k) \quad (2.6)$$

$$P(k+1|k) = A(k)P(k|k)A(k)^T + Q(k) \quad (2.7)$$

De seguida, faz-se uma estimativa do valor medido a partir do estado previsto (equação (2.8)) e é efetuada uma medição $z(k+1)$ e o filtro passa para o estado de correção de estado, onde é calculado o ganho do filtro (equação (2.9)), e são atualizadas a estimativa do estado com a medida $z(k+1)$ (equação (2.10)) e a covariância do erro da estimativa de estado (equação (2.11)).

$$\hat{y}(k+1|k) = C(k+1)\hat{x}(k+1|k) \quad (2.8)$$

$$K(k+1) = P(k+1|k)C(k+1)^T [C(k+1)P(k+1|k)C(k+1)^T + R]^{-1} \quad (2.9)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)[z(k+1) - \hat{y}(k+1|k)] \quad (2.10)$$

$$P(k+1|k+1) = [I - K(k+1)C(k+1)]P(k+1|k) \quad (2.11)$$

Como as funções densidade de probabilidade são gaussianas, pode-se definir uma região com a forma de uma elipse que é o lugar geométrico com probabilidade maior do que um determinado valor em torno do estado previsto e estimado, relacionada com a covariância do erro da estimativa do estado [50–52].

2.4.1 Filtro de Kalman Estendido

Quando a dinâmica de um sistema ou a equação de saída da equação 2.4 é não linear, o filtro não linear ótimo tem de propagar a função densidade de probabilidade, tornando o processo computacionalmente pesado.

O Filtro de Kalman Estendido (EKF) é um filtro que, em vez de propagar a função Gaussiana (não linear), lineariza em cada ciclo as não linearidades do sistema em torno das últimas estimativas previstas e corrigidas para o estado, aplicado de seguida o filtro de Kalman ao sistema linearizado. A figura 2.3 apresenta uma comparação da dinâmica deste filtro em relação ao filtro de Kalman convencional.

Considerando duas função não lineares $f(\cdot)$ e $h(\cdot)$, os modelos do sistema e da observação ficam representados por:

$$\begin{aligned} x(k+1) &= f(x(k)) + w(k) \\ y(k) &= h(x(k)) + v(k) \end{aligned} \quad (2.12)$$

sendo $w(k)$ o ruído de estado e $v(k)$ o ruído de medição.

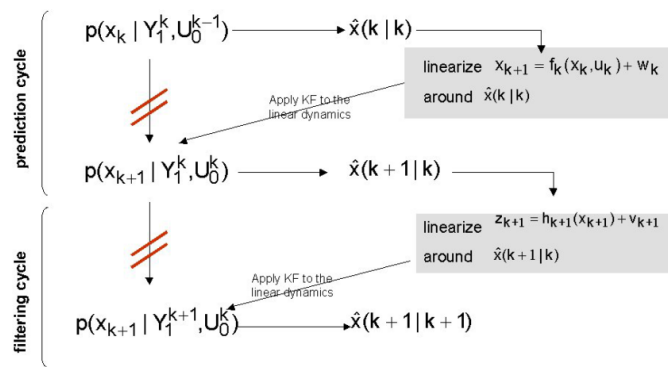


Figura 2.3: Dinâmica do Filtro de Kalman Estendido [51]

As equações da fase de predição são dadas por:

$$\begin{aligned}\hat{x}(k+1|k) &= f(\hat{x}(k|k)) \\ P(k+1|k) &= F(k)P(k|k)F^T(k) + Q(k)\end{aligned}\quad (2.13)$$

E as equações da fase de correção dadas por:

$$\begin{aligned}\hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K(k+1)[z(k+1) - h(k+1)(\hat{x}(k+1|k))] \\ K(k+1) &= P(k+1|k)H^T(k+1)[H(k+1)P(k+1|k)H^T(k+1) + R(k+1)]^{-1} \\ P(k+1|k+1) &= [I - K(k+1)H(k+1)]P(k+1|k)\end{aligned}\quad (2.14)$$

onde $F(k)$ e $H(k)$ são os jacobianos de $f(\cdot)$ e $h(\cdot)$ [51]:

$$\begin{aligned}F(k) &= \nabla f|_{\hat{x}(k|k)} \\ H(k+1) &= \nabla h|_{\hat{x}(k+1|k)}\end{aligned}\quad (2.15)$$

2.4.2 Unscented Kalman Filter

Na secção 2.4.1 é apresentado o princípio de funcionamento de um Filtro de Kalman Estendido (EKF). Contudo, esta abordagem poderá ter alguns problemas, visto que ao aproximar o estado por uma distribuição de probabilidade gaussiana que é posteriormente linearizada em cada ciclo pode introduzir erros no valor médio e na covariância do erro [53], sendo apenas viáveis em sistemas que são quase lineares em certos intervalos temporais, ou onde a diferença entre o estado estimado e o estado real é muito pequena [54].

O *Unscented Kalman Filter* apresenta uma solução para os problemas de aproximações do Filtro de Kalman Estendido. Este estimador tem por base a *Unscented Transform* (UT), usa um conjunto de pontos (denominados de *sigma points*) que parametrizam a média e covariância, conseguindo-se obter um desempenho semelhante a um filtro de Kalman convencional sem necessidade de linearizar o sistema.

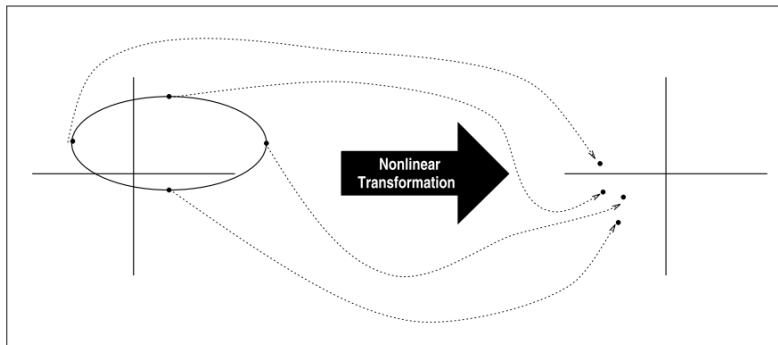


Figura 2.4: Princípio da (*unscented transform*) [54]

Uma variável x de dimensão n , média \bar{x} e covariância P_{xx} por um conjunto de $2n + 1$ pontos com pesos W_i .

$$\begin{aligned} X_0 &= \bar{x} & W_0 &= k/(n+k) \\ X_i &= \bar{x} + (\sqrt{(n+k)P_{xx}})_i & W_i &= 1/2(n+k) \\ X_{i+n} &= \bar{x} - (\sqrt{(n+k)P_{xx}})_i & W_{i+n} &= 1/2(n+k) \end{aligned} \quad (2.16)$$

onde $k \in \mathbb{R}$, $(\sqrt{(n+k)P_{xx}})_i$ é a i -ésima linha ou coluna na matriz raiz $\sqrt{(n+k)P_{xx}}$ e W_i é o peso associado ao i -ésimo ponto.

De seguida, é aplicada uma função não linear a estes pontos, resultando numa nuvem de pontos transformados de média \bar{y} e covariância P_{yy} .

$$\begin{aligned} y_i &= f[X_i] \\ \bar{y} &= \sum_{i=0}^{2n} W_i y_i \\ P_{yy} &= \sum_{i=0}^{2n} W_i \{y_i - \bar{y}\} \{y_i - \bar{y}\}^T \end{aligned} \quad (2.17)$$

A fase de previsão do *Unscented Kalman Filter* pode ser dividida em três fases. Na fase de previsão, calculam-se o conjunto de *sigma points* dados pela equação (2.16), estes são transformados segundo o modelo do processo e estima-se o novo estado $\hat{x}(k+1|k)$ e a covariância associada $P(k+1|k)$

$$\begin{aligned} X_i(k+1|k) &= f[X_i^a(k|k), u(k), k] \\ \hat{x}(k+1|k) &= \sum_{i=0}^{2n^a} W_i X_i^a(k+1|k) \\ P(k+1|k) &= \sum_{i=0}^{2n^a} W_i \{X_i(k+1|k) - \hat{x}(k+1|k)\} \{X_i(k+1|k) - \hat{x}(k+1|k)\}^T \end{aligned} \quad (2.18)$$

Numa segunda fase, prevê-se a observação esperada $\hat{z}(k+1|k)$ e covariância $P_{vv}(k+1|k)$:

$$\begin{aligned} Z_i(k+1|k) &= h[X_i(k+1|k), u(k), k] \\ \hat{z}(k+1|k) &= \sum_{i=1}^{2n^a} W_i Z_i(k+1|k) \\ P_{vv} &= R(k+1) + \sum_{i=0}^{2n^a} W_i \{Z_i(k+1|k) - \hat{z}(k+1|k)\} \{Z_i(k+1|k) - \hat{z}(k+1|k)\}^T \end{aligned} \quad (2.19)$$

Por fim, calcula-se a matriz de correlação cruzada $P_{xz}(k+1|k)$:

$$P_{xz}(k+1|k) = \sum_{i=0}^{2n^a} W_i \{X_i(k+1|k) - \hat{x}(k+1|k)\} \{Z_i(k+1|k) - \hat{z}(k+1|k)\}^T \quad (2.20)$$

De seguida, pode-se rescrever o vetor de estado como na equação (2.21) e o modelo de processo como na equação (2.22).

$$x^a(k) = \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \quad (2.21)$$

$$x(k+1) = f[x^a(k), u(k), k] \quad (2.22)$$

A transformada de incerteza usa $2n^a + 1$ *sigma points* dados por:

$$\begin{aligned} \hat{x}^a(k|k) &= \begin{pmatrix} \hat{x}(k|k) \\ 0_{q \times 1} \end{pmatrix} \\ P^a(k|k) &= \begin{bmatrix} P(k|k) & P_{xv}(k|k) \\ P_{xv}(k|k) & Q(k) \end{bmatrix} \end{aligned} \quad (2.23)$$

Desta forma, consegue-se obter com precisão a média e a covariância sem ser necessário resolver as matrizes Jacobianas e Hessianas do sistema [54].

Capítulo 3

Caracterização do Problema

No presente capítulo será definido o problema que deu origem a esta dissertação, assim como a solução proposta para a mesma.

3.1 Definição do Problema

A crescente automatização de processos tem levado a um crescimento acelerado nas áreas da robótica e inteligência artificial. Inicialmente, era adotada uma arquitetura centralizada com um número limitado de agentes em uso. Contudo, o desenvolvimento tecnológico levou a que fosse possível ter mais robôs a operar em simultâneo, com capacidade de realizar tarefas mais complexas cumprindo requisitos temporais mais apertados. A cooperação entre agentes torna-se, assim, num requisito imprescindível, levando à necessidade da existência de um supervisor. Este deve ser robusto, ter a capacidade de fundir a informação obtida por cada agente, de adaptar a tática da equipa de acordo com as condições de operação em cada instante de tempo e de suportar um número variável de agentes em serviço. O problema em questão está presente nas diversas áreas de robótica, como a busca e salvamento, vigilância e mapeamento de ambientes. Embora haja ligeiras diferenças, a arquitetura principal e a filosofia de trabalho é partilhada pelas diversas aplicações.

Para além disso, a capacidade de um agente conseguir seguir um determinado objeto de interesse é cada vez mais um requisito nas mais diversas aplicações robóticas. Porém, o robô nem sempre está equipado com os sensores necessários para a medição das grandezas que se pretende medir, tornando-se, assim, necessário estimar essas grandezas a partir de outras cuja observação seja possível. Da mesma forma, as observações realizadas pelos agentes estão sujeitas a ruído que pode ser prejudicial para a aplicação, ou o próprio objeto pode sair fora do campo de visão do agente que o está a seguir. Assim, para garantir uma boa qualidade de *tracking* é necessário recorrer a técnicas que permitam não só filtrar ruído, como também estimar as grandezas em análise no caso em que não existe observação (momentânea ou sistematicamente).

Esta dissertação tem como objetivo o desenvolvimento de um supervisor que contenha as características anteriormente descritas, sendo capaz de coordenar e fundir a informação de uma

equipa de robôs de modo atingirem o objetivo pretendido. Deseja-se ainda que cada agente seja dotado da capacidade de obter uma boa estimativa de posição e velocidade de um objeto de interesse.

3.2 Solução Proposta

Como solução ao problema apresentado, propõe-se o desenvolvimento de um supervisor escrito numa linguagem de programação de alto nível, que contenha uma estrutura hierárquica de controlo, com clara distinção dos cinco níveis de planeamento e controlo (táticas, jogadas, papéis, tarefas e ações básicas). Como exemplo de aplicação, foi realizado um treinador de futebol robótico, devido à possibilidade de se realizarem testes em ambiente de simulação, assim como em ambiente real com os robôs da equipa de futebol MSL 5DPO. Embora o supervisor seja orientado para a sua aplicação (a nível de táticas, papéis e tarefas), este tem as características presentes num supervisor para qualquer outra aplicação, tais como a sua estrutura e capacidade de trabalhar com um número variável de agentes.

Para além disso, foram desenvolvidos dois mecanismos de fusão sensorial. O primeiro, implementado no treinador, teve como objetivo a estimação de grandezas como a posição e velocidade da bola tendo em conta a observação de cada jogador. Esta fusão é essencial, na medida que a posição da bola influencia a tática e jogada da equipa, assim como os papéis e tarefas atribuídos a cada agente. O segundo foi implementado no *software* de decisão que corre em cada agente. Este teve como objetivo a estimação de grandezas como a velocidade e posição da bola, tendo em conta apenas a observação feita por esse robô. Em ambos os casos foram utilizadas e comparadas algumas ferramentas de fusão robustas e computacionalmente leves como, por exemplo, o filtro de Kalman e algumas heurísticas.

Capítulo 4

Equipa 5DPO

Este capítulo retrata a equipa de futebol robótico 5DPO. Começarão por ser apresentadas a constituição e o historial da equipa, seguindo-se duas secções acerca do *hardware* e arquitetura do *software* da mesma. A primeira pretende dar a conhecer os robôs utilizados nos testes, enquanto que a segunda contém a arquitetura em que esta dissertação se inseriu.

4.1 Contexto

5DPO é a equipa de futebol robótico da FEUP e do CRIIS - Centro de Robótica Industrial e Sistemas Inteligentes do INESC TEC. Esta é constituída por um grupo de professores da Faculdade de Engenharia da Universidade do Porto, investigadores do CRIIS, e estudantes do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores da FEUP.

A equipa conta com várias participações na liga MSL do Festival Nacional de Robótica, tendo já participado na RoboCup nas categorias de SSL (*Small Size League*), MSL (*Middle Size League*) e *Humanoid*.

Esta atividade visa desenvolver competências em várias áreas de investigação, tais como visão artificial, localização, fusão sensorial, controlo em tempo real, decisão e cooperação de equipas de robôs móveis [55].

4.2 Hardware

A equipa MSL 5DPO conta com seis robôs móveis que respeitam as restrições apresentadas na secção 2.3.1.

Os robôs são dotados de tração omnidirecional de 3 rodas, o que lhes permite realizar movimentos simples ou combinados em qualquer direção. O mecanismo de *kicker* baseia-se em princípios eletromagnéticos. Este é constituído por um solenóide que, durante os instantes de *kick*, é alimentado com 100V, através de uma bateria de condensadores. Um segundo solenóide dá a opção de realizar quer passes/remates rasteiros, quer *lob shots*.

À semelhança da maioria das equipas desta liga, os robôs da equipa 5DPO contam com um sistema de visão catadióptrico. Como é visível na figura 4.1b, este sistema baseia-se numa câmara industrial apontada para um espelho convexo, dotando o robô de um ângulo de visão de 360°. Através de simples equações é possível fazer o mapeamento de pontos da imagem capturada no plano xOy . Para além disso, os robôs estão equipados com uma bússola digital, que permite ajudar na localização, bem como com um sensor ótico para detetar a bola.

Na figura 4.1a, estão representados os robôs da equipa 5DPO. Observando com detalhe, é visível que o robô 6 apresenta algumas diferenças em relação aos restantes. Estas são fruto do desenvolvimento de nova eletrónica, assim como um novo mecanismo de *grabber*.



(a) Foto de equipa

(b) Sistema de visão

Figura 4.1: Robôs da equipa *Middle Size League* 5DPO

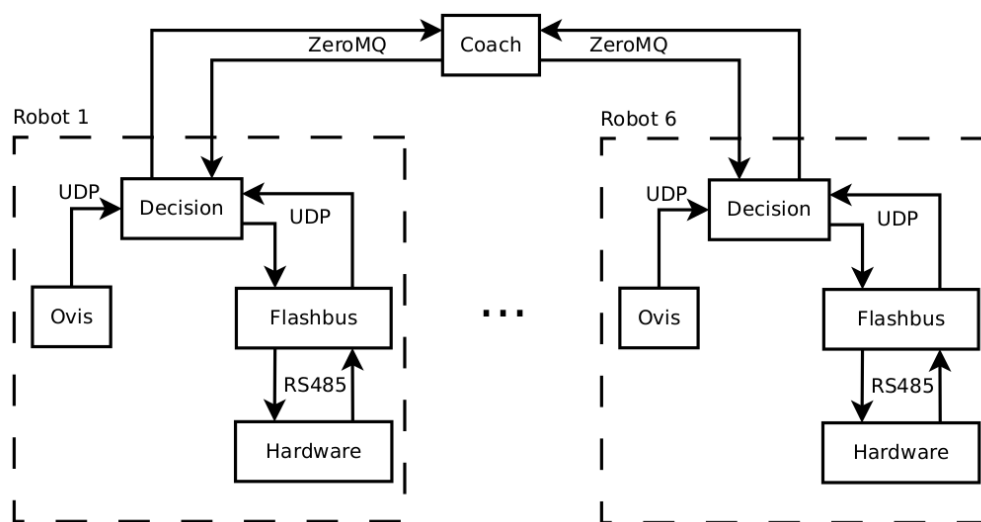
4.3 Arquitetura do Software

Um jogo de futebol robótico envolve a cooperação de múltiplos agentes. O conhecimento e interação com os restantes companheiros de equipa torna-se, assim, um fator determinante para se obter sucesso neste desporto.

Em resposta a este desafio, a equipa 5DPO apresenta a arquitetura presente na figura 4.2. Os robôs são completamente autónomos, tomando as decisões das ações a executar em função do seu papel internamente, no *Decision*. Um supervisor externo é responsável por garantir a cooperação entre os agentes (*Coach*). Nas secções seguintes, serão apresentados os nós mais importantes da arquitetura da equipa.

4.3.1 Flashbus

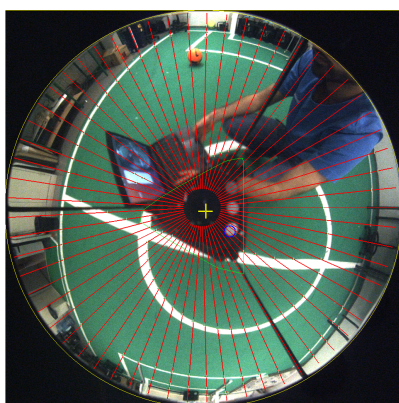
O *Flashbus* é o protocolo utilizado para as comunicações série entre o *software* de decisão (*Decision*) e os microcontroladores presentes nas placas de *hardware* do robô. As comunicações são baseadas no modelo mestre-escravo, havendo uma aplicação mestre que corre localmente em cada agente (também denominada por *Flashbus*). Esta é responsável pela leitura e escrita dos sensores/atuadores.

Figura 4.2: Arquitetura de *software* da equipa 5DPO

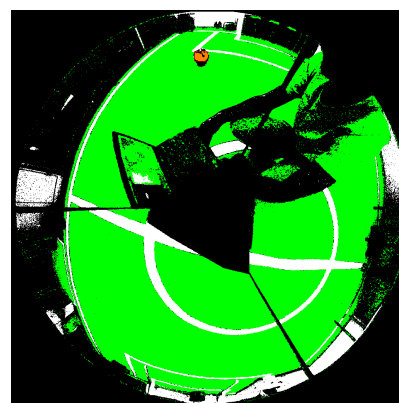
4.3.2 Ovis

O *Big Omni Vision* é o programa responsável pela aquisição e processamento de imagem.

O principal objetivo deste *software* é conseguir detetar as linhas do campo e objetos de interesse, como bola e obstáculos (atualmente apenas é detetada a bola). Consultando as regras da competição, verifica-se que as linhas são marcações brancas sobre fundo verde e, tipicamente, a bola costuma ter uma cor bem definida. Desta forma, os *clusters* a detetar podem ser identificados através da sua cor. Para se identificar as linhas, procede-se à deteção de transições verde-branco em linhas com direção radial, a partir do centro da imagem. São também identificados um conjunto de bolas que cumprem o requisito da sua cor e tamanho, atribuindo uma qualidade às mesmas de acordo com a relação do número de pixels que o *blob* contém e aqueles que um *blob* numa bola àquela distância deveria ter.



(a) Imagem não processada



(b) Imagem processada

Figura 4.3: Imagem capturada pelo sistema de visão

4.3.3 Coach

O *Coach* é um supervisor de alto nível, responsável por garantir a cooperação dos agentes em campo.

Este programa corre na *Base Station* da equipa, estando responsável por interpretar os comandos da *Referee Box* e traduzir os mesmos em alterações na tática e jogada da equipa. Juntando estas com a informação vinda de cada agente, o treinador define qual o melhor papel a desempenhar por cada jogador.

Uma descrição mais detalhada do treinador será feita nos capítulos 7 e 8.

4.3.4 Decision

O *Decision* é o programa *core* de todo o sistema, sendo responsável por decidir o comportamento que o robô vai desempenhar. Em função do papel que o treinador lhe atribuiu, o *Decision* vai executar uma sequência de tarefas, que por sua vez definem quais as ações que o robô vai executar.

Um dos fatores primordiais para um bom desempenho em campo é a capacidade do agente conseguir construir uma boa representação do mundo que o rodeia. Nesse sentido, o *Decision* usa informação da visão, vinda do *Ovis* e dos restantes sensores, obtida por *Flashbus*, para construir o seu *worldstate*. Para a localização, procede-se à fusão da informação vinda da odometria, visão (linhas) e bússola digital, através de um filtro de Kalman extendido. A estimação de posição e velocidade da bola são realizados localmente através dum filtro de Kalman. As trajetórias são geradas com base num algoritmo A* adaptado do utilizado pela geração SSL [56], permitindo o desvio de obstáculos (no caso de serem detetados), e facilitando manobras de aproximação à bola.

4.4 Arquitetura de Cooperação

A arquitetura base do ecossistema da equipa 5DPO assenta numa estrutura hierárquica, podendo-se claramente distinguir cinco níveis: táticas (*tactics*), jogadas (*plays*), papéis (*roles*), tarefas (*tasks*) e ações (*actions*). As táticas correspondem ao objetivo comum que a equipa tenta alcançar (atacar, defender, etc). As jogadas expressam a situação do jogo num determinado instante de tempo (falta a favor dos adversário, lançamento para a equipa, jogo normal, etc). Os papéis representam as funções desempenhadas pelos agentes (atacante, defesa, médio, etc), caracterizando o seu comportamento em campo. As tarefas são as habilidades desempenhadas por um robô (por exemplo, ir para o ponto x , driblar, fazer um passe, etc). Finalmente, as ações especificam as atuações de baixo nível desempenhadas pelos agentes (*GoToXYTheta*, *Rotate*, *Stop*, etc) [57]. Esta estrutura foi baseada no trabalho desenvolvido por Reis *et al.* [24].

Na verdade, antes do início desta dissertação a equipa havia esquecido o conceito de tática, sendo que o treinador até então existente apenas se baseava na jogada para a atribuição dos papéis aos diversos agentes.

Devido à robustez e eficiência demonstradas por esta estrutura em anos anteriores, decidiu-se manter a mesma, reintroduzindo o conceito de tática, de forma a obter maior flexibilidade na atribuição de papéis aos robôs (como será analisado no capítulo 7). A figura 4.4 contém a arquitetura de cooperação atual da equipa 5DPO. As táticas e jogadas são definidos e atualizados pelo treinador, e são conceitos comuns a toda a equipa. Por sua vez, os papéis, tarefas e ações são executados individualmente pelos agentes em campo.

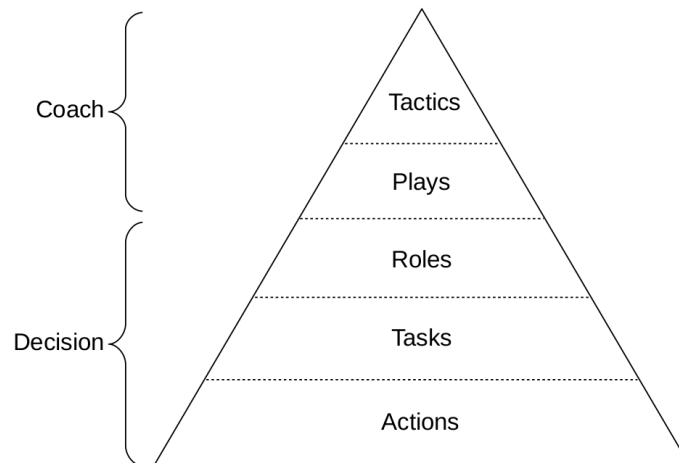


Figura 4.4: Arquitetura Cooperação

Durante o período de jogo, o treinador constrói uma representação do mundo através da percepção realizada pelos agentes ativos da equipa. Ao mesmo tempo, o treinador encontra-se ligado à *Referee Box*, permitindo que este adquira uma noção da situação do jogo em cada instante de tempo.

Capítulo 5

Ambiente de Simulação

Neste capítulo, será apresentado o ambiente de simulação utilizado para alguns testes desta dissertação. Desta forma, começará por ser apresentado o simulador SimTwo em geral, seguindo-se uma secção que particulariza o cenário de futebol robótico desenvolvido pela equipa 5DPO.

5.1 SimTwo

O SimTwo ¹ é um ambiente de simulação robótico 3D bastante realista, capaz de simular a cinemática e dinâmica de vários tipos de robôs.

Este simulador baseia-se cenários que podem conter vários objetos do tipo de *robots*, *obstacles* e *things*. Um robô é definido por um conjunto de corpos rígidos interligados por juntas (rotacionais, esféricas ou prismáticas), tendo em conta as propriedades físicas dos mesmos. A utilização do motor de simulação ODE ² torna a dinâmica deste simulador realista, considerando características como a massa, forma, momento de inércia, coeficientes de atrito e elasticidade dos corpos. Este programa está ainda dotado de um conjunto de sensores e atuadores que permitem que o robô percecione e interaja com o mundo à sua volta. Um robô pode conter ainda elementos do tipo *shell*, que consistem em sólidos cujas propriedades físicas não são tidas em conta, sendo apenas usados em colisões ou para modificar o aspeto exterior do robô. Os *obstacles* e *things* são compostos por sólidos e diferenciam-se em situações de colisão, em que, enquanto que os primeiros permanecem imóveis, os segundos reagem de acordo com as suas propriedades físicas.

Uma das características do SimTwo é a possibilidade de programação de um *script* de controlo em Pascal. Esta funcionalidade é implementada através da biblioteca Pascal Script e, permite controlar não só o comportamento dos robôs, como também dos restantes elementos do cenário.

O SimTwo foi desenvolvido em Object Pascal, corre em Windows (ou Linux, via wine³) e é gratuito. Embora seja um *software* de código fechado, permite a criação de novos cenários, possibilitando a simulação de qualquer tipo de robô. Para além disso, este simulador fornece

¹<http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo>

²<http://www.ode.org>

³<https://www.winehq.org>

mecanismos de comunicação série com microcontroladores e por rede para interligação com outras aplicações [58, 59].

5.2 Simulador MSL 5DPO

Num projeto como o futebol robótico, é crucial ter uma plataforma onde seja possível realizar testes, sem necessidade de utilizar robôs reais. Numa fase inicial do desenvolvimento, a utilização de um simulador permite testar algoritmos de alto nível de uma forma mais rápida e num ambiente mais controlado do que uma aplicação imediata em robôs reais. Para a realização de testes nesta dissertação, procurava-se um simulador que fosse realista, facilmente configurável, permitisse interação com uma aplicação externa e que permitisse representar os robôs da equipa da forma mais aproximada possível. Constatou-se que o SimTwo cumpria todos os requisitos, pelo que se optou pela sua utilização.

Na verdade, equipa já apresentava uma primeira versão de um cenário para este simulador (figura 5.1a). Contudo, esta carecia de alguns sensores e atuadores presentes nos robôs e o seu *script* de controlo estava bastante incompleto. Desta forma, procedeu-se ao desenvolvimento desta versão, adicionando o mecanismo de *grabber* presente na maioria dos robôs da equipa, um sensor de bola e a funcionalidade de *low kick*. Ainda ao nível de modelação 3D, foram feitas outras pequenas modificações que tornaram o simulador computacionalmente mais leve (permitindo ter um maior número de robôs a operar ao mesmo tempo), e que replicavam um controlo da bola semelhante ao dos robôs reais da equipa. O modelo de dinâmica do robô já tinha sido anteriormente estudado e desenvolvido por Caldas [60], Conceição *et al.* [61] e Nascimento [62], pelo que se optou por adotar o mesmo modelo. A figura 5.1b apresenta a versão final da representação do robô da equipa 5DPO em SimTwo.

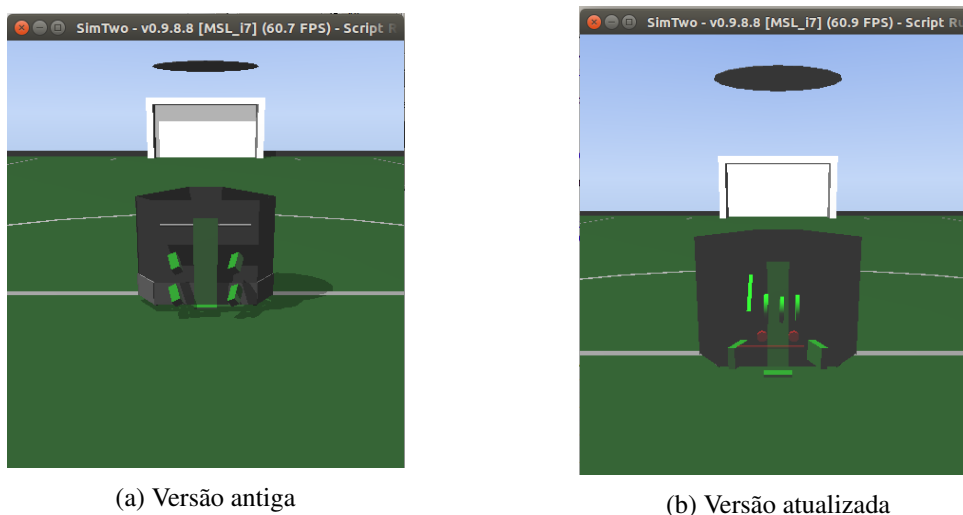


Figura 5.1: Representação dos robôs da equipa MSL 5DPO no ambiente de simulação SimTwo

O *script* de controlo foi totalmente refeito, de acordo com a estrutura presente na figura 5.3. O seu tempo de ciclo é de 40ms, e foi escolhido de modo a coincidir com o tempo de ciclo do

software de visão da equipa (Ovis). O *loop* principal de controlo pode ser analisado com maior detalhe na figura 5.2.

O cenário principal contém um campo com as dimensões oficiais, e é constituído por seis robôs da equipa 5DPO e cinco adversários. Para facilitar a interação com o utilizador, desenvolveu-se uma interface gráfica com o mesmo, na janela *Sheets* (figura 5.5). Nesta é possível monitorizar o estado do mundo (nomeadamente as poses dos robôs, estado dos seus sensores e atuadores e percepção da bola no seu referencial local), visualizar e forçar a posição da bola no referencial do mundo, ativar/desativar o ruído na observação da bola, mudar a potência máxima do *kicker*, mudar o modo de funcionamento, ativar/desativar os obstáculos e, finalmente, repor as posições dos diferentes constituintes do cenário para os seus valores por defeito. O modo de funcionamento consiste em controlo manual (*keys*), ou controlo via aplicação externa (*decision*), por comunicações UDP. A figura 5.4 contém uma representação do cenário de simulação completo.

Para simular o comportamento dos adversários, começou-se por se adicionar ao cenário mais cinco robôs. No entanto, verificou-se uma queda acentuada da capacidade de simulação em tempo real devido à elevada carga computacional requerida pela biblioteca de simulação da dinâmica dos corpos (ODE), tornando, assim, a simulação impraticável. Por conseguinte, representaram-se os adversários com cinco cilindros modelizados como *things*. Estes podiam-se aleatoriamente no interior do campo, à exceção do guarda-redes, cujo movimento estava restrito à pequena área.

5.2.1 Conclusões

A continuação do desenvolvimento do cenário deste simulador aplicado à equipa *Middle Size* 5DPO foi fulcral para o trabalho realizado nesta dissertação. Desta forma, obteve-se uma plataforma que simula não só cinemática mas também dinâmica de robôs, capaz de suportar os testes dos estimadores de posição e velocidade de bola desenvolvidos para cada robô, assim como os testes de cooperação entre os diversos agentes da equipa. Além disso, o SimTwo foi uma ferramenta de elevada utilidade no Robótica 2017⁴, permitindo testar a implementação dos *roles* de cada agente e a cooperação dos mesmos nos períodos em que o campo se encontrava ocupado com jogos, ou com testes por parte de outras equipas.

⁴<http://robotica2017.isr.uc.pt/index.php/pt/>

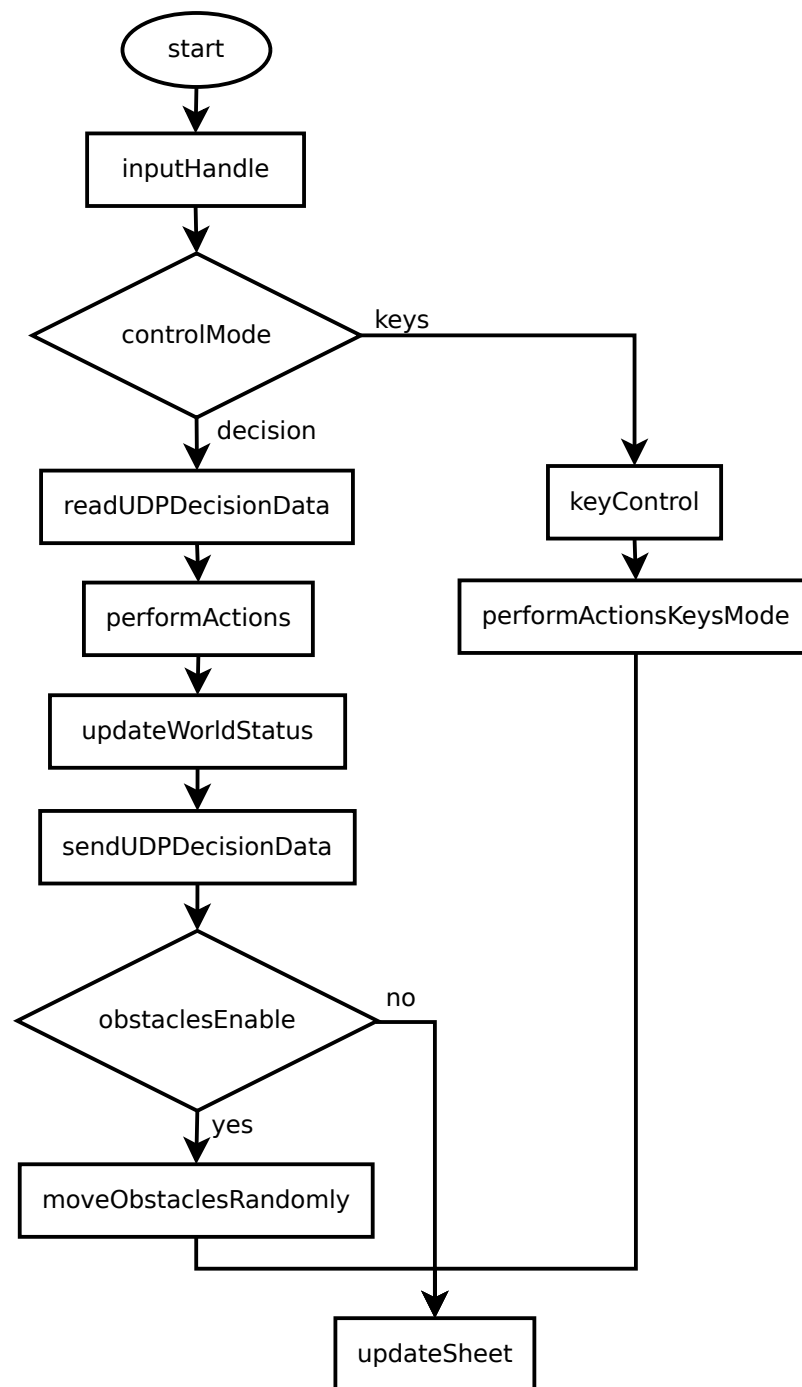


Figura 5.2: Fluxograma do ciclo principal do *script* de controlo do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo

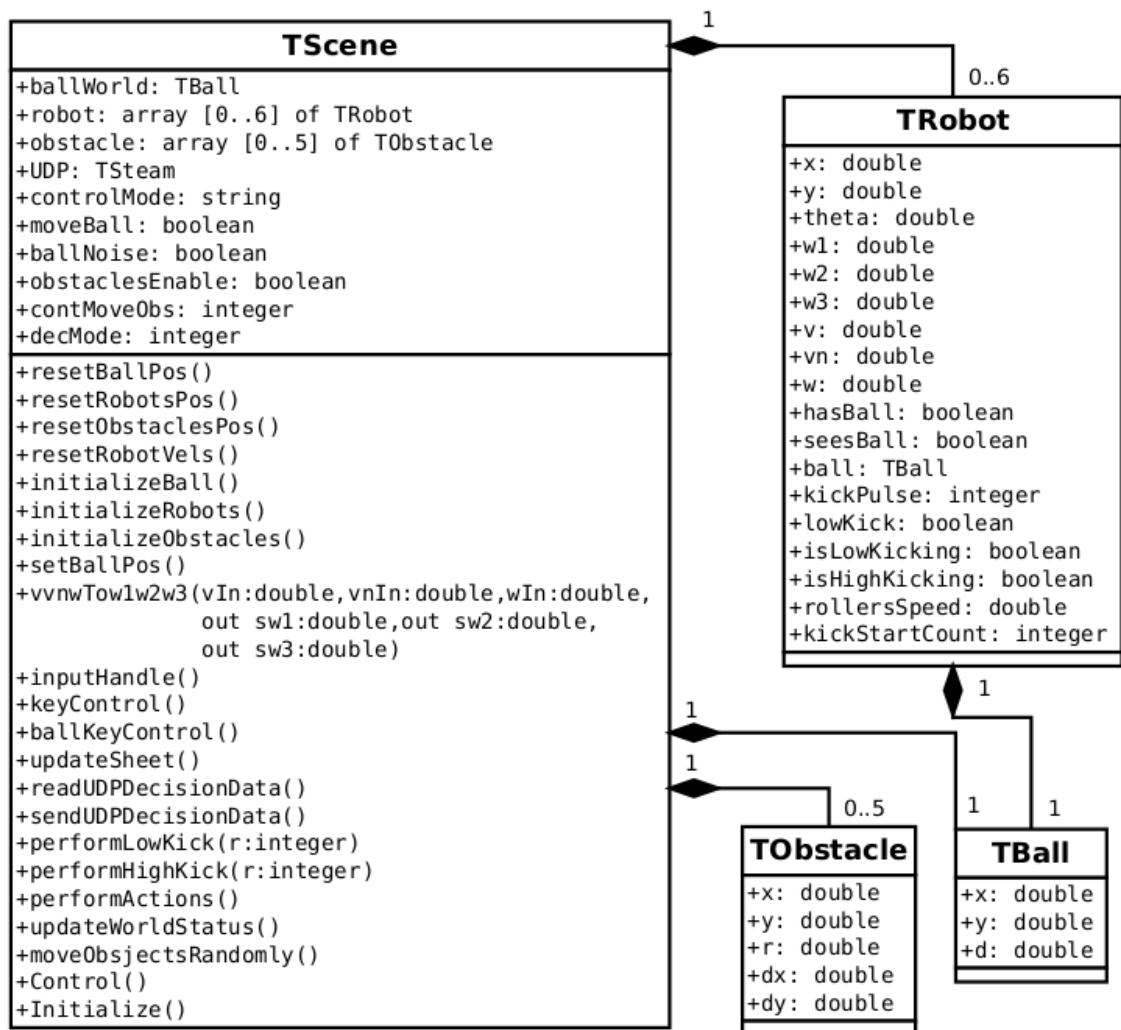


Figura 5.3: Diagrama UML de classes do *script* de controlo do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo



Figura 5.4: Janela principal do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo

Sheets																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	robot	posX	posY	posTheta	hasBall	seesBall	w1	w2	w3	lowKick	y	vn	w	ballSens	kickerCount	rollers	kickPulse	ballXrob	ballYRob	ballDrob
2	0	-3	-1.69E-00	3.16E-005	false	true	0	0	0	false	0	0	0	-1	0	-1.2E004	0	9	0.000376	9
3	1	-6	-5.49E-00	-5.03E-00	false	true	0	0	0	false	0	0	0	-1	0	0	0	6	0.000503	6
4	2	-3	-1.88E-00	-2.08E-00	false	true	0	0	0	false	0	0	0	-1	0	0	0	3	-0.00099	3
5	3	-6	-3	0.4	false	true	0	0	0	false	0	0	0	-1	0	0	0	6.7	0.426	6.71
6	4	-6	3	-0.4	false	true	0	0	0	false	0	0	0	-1	0	0	0	6.7	-0.426	6.7
7	5	-1	-6.3	1.54	false	true	0	0	0	false	0	0	0	-1	0	0	0	6.33	-0.805	6.38
8																				
9																				
10					noise	toggle			obstacles	toogle			DEC modu	toogle						
11					true				false				centralize							
12	resetObs																			
13	resetBall	resetRob	resetAll		Robot	ContMode	MoveBall	toggle			x	y	forceBall							
14	keys	decision	rollersIn		1	decision	false				0	0	set							
15	lowKick	highKick	rollersOut																	
16					in kickPulse						ballPos	xw	yw							
17					40						0	0								
18	keys	move robot																		
19	A	moveBall Left																		
20	S	moveBall Down																		
21	D	moveBall Right																		
22	W	moveBall Up																		
23	C	w +																		
24	Z	w -																		
25	O	lowKick																		
26	E	highKick																		
27	n	rollersIn enable																		
4																				

Figura 5.5: Janela *Sheets* do cenário de futebol robótico MSL 5DPO no ambiente de simulação SimTwo

Capítulo 6

Estimação de Posição e Velocidade da Bola por Parte de um Agente

O futebol é um desporto que se desenrola em torno de um elemento principal - a bola. Na verdade, quase todos os *roles* dependem da posição da bola, desde o defesa que se coloca na linha entre a bola e a baliza, o médio que vai fazer ou receber um passe, até ao atacante que vai de encontro à bola para rematar à baliza. Desta forma, é fundamental obter uma estimativa precisa da localização da bola. Porém, esta tarefa não é trivial, visto que os robôs nem sempre conseguem ver a bola (porque esta saiu do seu campo de visão, ou porque outro robô se colocou à frente da mesma), vêm falsos positivos e as observações efetuadas estão sujeitas a ruído que é proporcional à distância da bola ao mesmo. Por conseguinte, é necessário realizar algum processamento para garantir a obtenção de uma estimativa de boa qualidade.

Para além disso, as manobras de aproximação e receção de bola são cruciais para o sucesso de uma jogada. O conhecimento do vetor de velocidade da bola permite adaptar a velocidade do robô ao mesmo, de forma a conseguir-se uma receção suave e sem ressaltos.

Este capítulo visa o estudo de diferentes técnicas de estimação de posição e velocidade da bola por parte de um agente. Começará por ser apresentado o sistema de observação da bola adotado pela equipa 5DPO, seguindo-se duas propostas de estimação da posição e velocidade da bola que recorrem a filtros de Kalman, concluindo com um estimador de velocidade baseado numa regressão linear através do método dos mínimos quadrados, à semelhança do que foi apresentado por Silva *et al.* [37, 43]. As duas implementações de filtros de Kalman diferem no referencial em que o vetor de estado se encontra referenciado. Enquanto que na primeira abordagem o vetor de estado encontra-se referenciado no referencial local do robô, na segunda encontra-se referenciado num referencial global (do mundo). Repare-se que na primeira abordagem o robô nem precisa de saber a sua localização global. Esta abordagem abre o leque de aplicações onde se pode utilizar esta técnica, para além do futebol robótico. Por exemplo, em qualquer caso em que um robô quer intercepar um alvo, independentemente de ter ou não a sua localização global ativa.

Por fim, serão apresentados um conjunto de testes realizados de forma a validar e comparar as diferentes abordagens implementadas.

6.1 Aquisição de medida da bola

Na liga MSL, a observação da bola está a cargo de cada jogador. Os agentes da equipa 5DPO adquirem a posição da bola a partir do seu sistema de visão. Tal como foi referido na secção 4.2, a visão consiste num sistema catadióptrico, onde uma câmara apontada para um espelho adquire imagens com um ângulo de visão de 360° . A partir de cada frame são extraídos *clusters* de objetos de interesse, como a bola e obstáculos e, são detetadas transições verde-branco sob linhas radiais. Os *clusters* são separados de acordo com uma gama de cores previamente definidas nas regras¹ da Robocup, e são caracterizados por dois ângulos ρ e α , tal como é visível na figura 6.1. A bola a identificar é esférica, de tamanho 5² e de qualquer cor que não seja branco, preto ou verde [20]. Tipicamente, costumam-se utilizar bolas de cor amarela ou laranja.

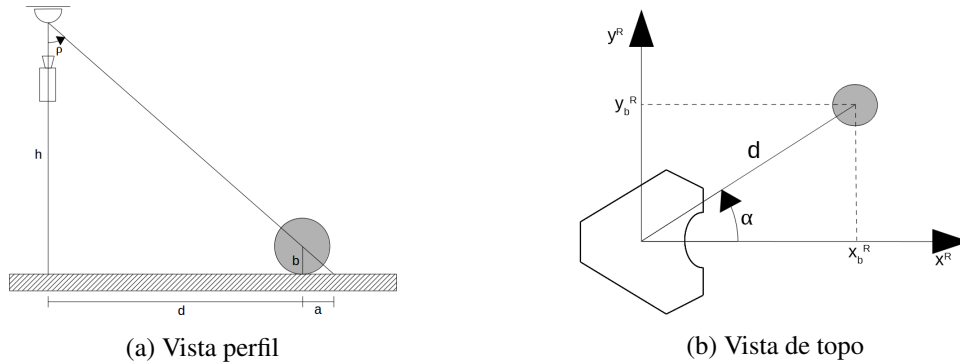


Figura 6.1: Sistema de visão

Tendo em conta a semelhança dos triângulos presentes na figura 6.1a, retira-se que:

$$\begin{aligned} \frac{b}{h} &= \frac{a}{a+d} \Leftrightarrow ab + db = ah \\ \Leftrightarrow a &= \frac{db}{h-b} \end{aligned} \quad (6.1)$$

Analisando a mesma figura e tendo em conta a noção trigonométrica de tangente, é possível obter a distância euclidiana da bola ao robô no plano 2D:

$$\tan(\rho) = \frac{d+a}{h} \Leftrightarrow d = h \tan(\rho) - a \quad (6.2)$$

Substituindo (6.1) em (6.2):

$$\begin{aligned} d &= h \tan(\rho) - \frac{db}{h-b} \Leftrightarrow dh - db = h \tan(\rho)(h-b) - db \\ \Leftrightarrow d &= \tan(\rho)(h-b) \end{aligned} \quad (6.3)$$

¹http://wiki.robocup.org/images/7/72/2017-msl-rules_v18_4.pdf

²de acordo com o *standard* da FIFA

Assim, a posição da bola no plano horizontal xOy é dada pelo par de coordenadas:

$$\begin{aligned}x_b^R &= d \cos(\alpha) \\ y_b^R &= d \sin(\alpha)\end{aligned}\tag{6.4}$$

6.2 Dinâmica da bola

Considere-se a bola como uma esfera, com massa m , que se desloca unicamente ao longo de uma superfície horizontal. É sabido que o movimento da bola pode ser descrito por duas fases distintas, nomeadamente, uma primeira em que apresenta um movimento de rotação com um dado deslizamento, e uma segunda em que apresenta apenas movimento de rotação, sem deslizamento [63].

Além desta e outras não linearidades, o facto de a bola estar inserida num ambiente extremamente dinâmico, onde está constantemente a ser transportada ou a sofrer ressaltos em jogadores, torna impossível a obtenção de um modelo que expresse a dinâmica da mesma com rigor.

Por esse motivo, considerou-se apenas que a bola apresentava um decaimento exponencial de velocidade no decorrer do tempo, dada por:

$$\vec{v}(k) = C\vec{v}(k-1)\tag{6.5}$$

Simplificando, suponha-se que nenhuma força externa está aplicada à bola. Considere-se ainda B_{eq} como atrito equivalente devido ao atrito viscoso e M_{eq} a massa equivalente devido ao momento linear e de rotação. Aplicando a segunda lei de Newton, obtém-se:

$$\sum F = M\dot{v}_x \Leftrightarrow M\dot{v}_x + B_{eq}v_x = 0 \Leftrightarrow \dot{v}_x = -C_1 v_x\tag{6.6}$$

onde

$$C_1 = \frac{B_{eq}}{M}\tag{6.7}$$

Considerando o vetor de estado (6.8), contendo a posição e velocidade da bola no referencial global, obtém-se o modelo de espaço de estados presente em (6.9)

$$X(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \\ v_{x_b}(t) \\ v_{y_b}(t) \end{bmatrix}\tag{6.8}$$

$$\begin{bmatrix} v_{x_b}(t) \\ v_{y_b}(t) \\ \dot{v}_{x_b}(t) \\ \dot{v}_{y_b}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -C_1 & 0 \\ 0 & 0 & 0 & -C_1 \end{bmatrix} \begin{bmatrix} x_b(t) \\ y_b(t) \\ v_{x_b}(t) \\ v_{y_b}(t) \end{bmatrix}\tag{6.9}$$

onde

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -C_1 & 0 \\ 0 & 0 & 0 & -C_1 \end{bmatrix} \quad (6.10)$$

Utilizando a transformada inversa de Laplace (6.12) é possível determinar a matriz de transição de estado tal que:

$$X(k) = AX(k-1) \quad (6.11)$$

$$A = e^{A_c T} = \mathcal{L}^{-1}([sI - A_c]^{-1}) \quad (6.12)$$

$$[sI - A_c]^{-1} = \begin{bmatrix} s & 0 & -1 & 0 \\ 0 & s & 0 & -1 \\ 0 & 0 & s + C_1 & 0 \\ 0 & 0 & 0 & s + C_1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s} & 0 & \frac{1}{s(s+C_1)} & 0 \\ 0 & \frac{1}{s} & 0 & \frac{1}{s(s+C_1)} \\ 0 & 0 & \frac{1}{s+C_1} & 0 \\ 0 & 0 & 0 & \frac{1}{s+C_1} \end{bmatrix} \quad (6.13)$$

Pelo que

$$A = \mathcal{L}^{-1}([sI - A_c]) = \begin{bmatrix} 1 & 0 & \frac{1-e^{-C_1 T}}{C_1} & 0 \\ 0 & 1 & 0 & \frac{1-e^{-C_1 T}}{C_1} \\ 0 & 0 & e^{-C_1 T} & 0 \\ 0 & 0 & 0 & e^{-C_1 T} \end{bmatrix} \quad (6.14)$$

Assim, o modelo de transição de estado fica:

$$\begin{bmatrix} x_b(k) \\ y_b(k) \\ v_{x_b}(k) \\ v_{y_b}(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{1-e^{-C_1 T}}{C_1} & 0 \\ 0 & 1 & 0 & \frac{1-e^{-C_1 T}}{C_1} \\ 0 & 0 & e^{-C_1 T} & 0 \\ 0 & 0 & 0 & e^{-C_1 T} \end{bmatrix} \begin{bmatrix} x_b(k-1) \\ y_b(k-1) \\ v_{x_b}(k-1) \\ v_{y_b}(k-1) \end{bmatrix} \quad (6.15)$$

Defina-se $v_{x_b}^w(k)$ e $v_{y_b}^w(k)$ como as componentes x e y da velocidade da bola no instante k , no referencial global. A partir da expressão (6.15), considerando que C_1 é uma contante maior do que zero e $T = 40ms$ representa o período de amostragem, é visível que a velocidade decai com uma constante C (6.16), resultando em (6.17):

$$C = e^{-TC_1} \quad (6.16)$$

$$\begin{bmatrix} v_{x_b}^w(k) \\ v_{y_b}^w(k) \end{bmatrix} = C \begin{bmatrix} v_{x_b}^w(k-1) \\ v_{y_b}^w(k-1) \end{bmatrix} \quad (6.17)$$

Para determinar o valor da constante C , procedeu-se ao lançamento de uma bola em direção ao robô, com uma aceleração negativa, registrando-se a posição $x_{medido}(t)$ da mesma com intervalos de tempo de $40ms$. Durante toda a experiência o robô esteve parado. Supondo que a bola se

movimenta apenas segundo o eixo x , a trajetória descrita pela bola no ensaio pode ser estimada pela equação (6.18).

$$x_{est}(t) = x_f + (x_i - x_f)(1 - e^{-\frac{t}{C_1}}) \quad (6.18)$$

onde $x_{est}(t)$ corresponde à posição estimada no instante t , x_i e x_f correspondem às posições inicial e final da bola.

$$\sum_{i=0}^{t_{max}} (x_{est}(i) - x_{medido}(i))^2 \quad (6.19)$$

De seguida, com o auxílio de uma ferramenta de otimização numérica, procedeu-se à minimização da soma do erro quadrático dado pela equação (6.19), variando o valor de C_1 . Desta forma, obteve-se que $C_1 = 1.596$, com um somatório de erro quadrático igual a 1.909. A figura 6.2 contém as medidas da bola e as posições estimadas com o valor ótimo de C_1 .

Por fim, o valor de C pode ser calculado a partir da expressão (6.16), resultando que $C = 0.938$.

$$C = e^{-TC_1} = e^{-0.040 \times 1.596} = 0.938 \quad (6.20)$$

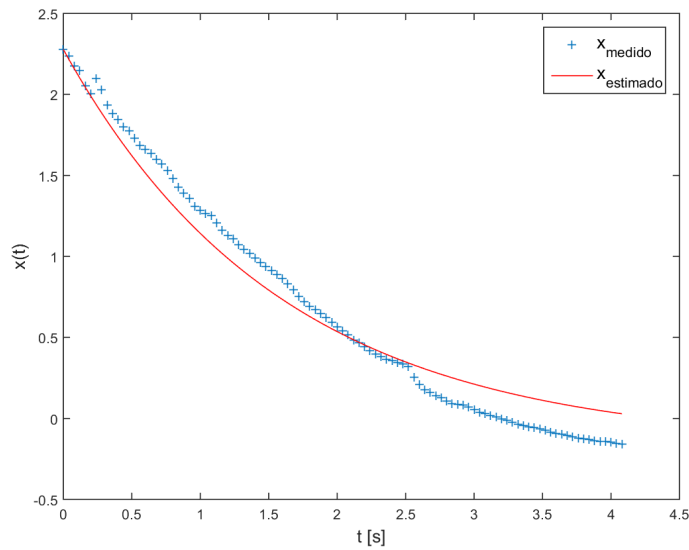


Figura 6.2: Comparação de x_{medido} e $x_{estimado}$

6.3 Filtro de Kalman

Como já foi referido no capítulo 2, o filtro de Kalman é uma ferramenta utilizada muito frequentemente no seguimento de objetos [15, 21, 37, 39, 42–45]. De facto, a sua robustez, simplicidade computacional e também a sua capacidade de incorporar o conhecimento que temos sobre a caracterização do ruído das observações, faz do filtro de Kalman uma ferramenta bastante útil, principalmente para aplicações em tempo real.

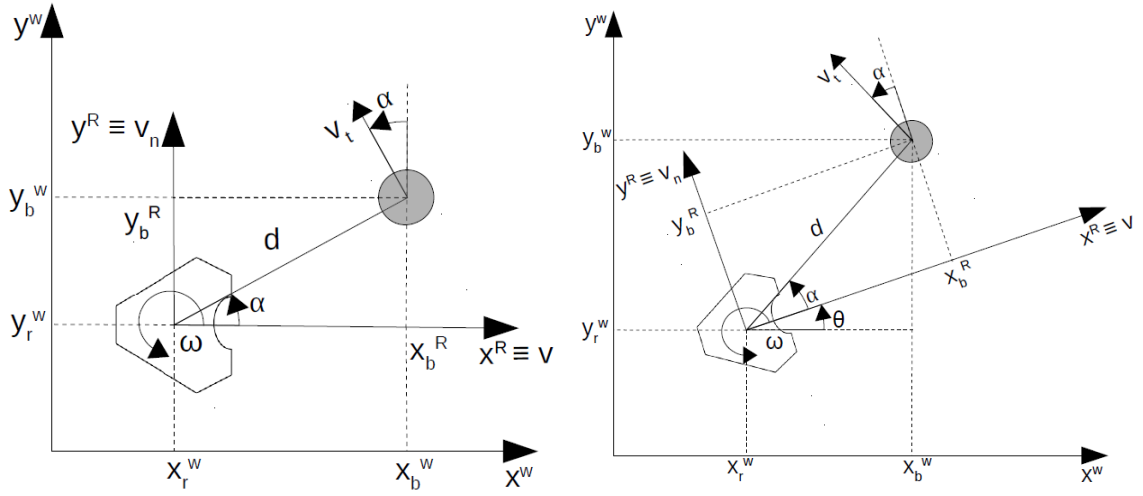
Perante o problema proposto, optou-se por se utilizar um filtro de Kalman. Desta forma, o filtro será utilizado não só para filtrar as medidas de posição da bola, como também para fornecer uma estimativa da velocidade da mesma.

A observação da bola é realizada no referencial local do robô e, a sua conversão para coordenadas no referencial global, do mundo, exige o conhecimento da pose do robô. Na verdade, os agentes têm uma dada incerteza associada à sua localização, pelo que as estimativas de posição da bola no referencial global serão afetadas pela mesma. Em determinadas ocasiões, os agentes precisam de saber a localização da bola no referencial global como, por exemplo, para comparar a sua estimativa de bola com a bola do treinador (que resulta da fusão das bolas de todos os agentes da equipa ativos). Contudo, se o robô não se encontrar bem localizado, ou até mesmo se não tiver localização, a sua perceção da bola no referencial global fica completamente comprometida, o que não acontece no caso de ter uma perceção referenciada a si mesmo. Para além disso, em circunstâncias como manobras de receção e/ou aproximações à bola, os robôs beneficiam da obtenção da posição e velocidade da bola no seu referencial local, obtendo, assim, resultados mais precisos, que não são prejudicados pela sua incerteza ou ausência, mesmo que momentânea, de uma estimativa da sua localização.

Desta forma, decidiu-se começar por estimar a posição e velocidade da bola através de um filtro de Kalman, com o seu vetor de estado referenciado no referencial local do robô (secção 6.4). Por forma a obter-se um elemento de comparação, considerou-se ainda o caso de o vetor de estado estar referenciado num referencial global (secção 6.5).

6.4 Filtro de Kalman - Referencial Local

Considere-se o sistema de coordenadas presente na figura 6.3. Tal como foi referido na secção 6.1, a observação da bola é realizada no referencial local do robô. Como este tem uma dada incerteza na sua localização, pretende-se utilizar variáveis que não dependam da posição x_r e y_r do robô, no referencial global, para a estimação de posição e velocidade da bola. Desta forma, são tomadas como entradas conhecidas as velocidades lineares do robô na direção v , vn e angular ω . Estas variáveis são calculadas a partir da estimativa do movimento do robô, nomeadamente, odometria, IMU, entre outros. Para além disso, a partir do *software* de visão, são conhecidas a distância euclidiana do centro da bola ao centro do robô no plano horizontal d , assim como o ângulo da bola no referencial local do robô α .



(a) Considerando referencial do robô alinhado com o referencial global
 (b) Considerando referencial do robô não alinhado com o referencial global

Figura 6.3: Sistema de coordenadas da observação da bola

Supondo um referencial global de inércia com a mesma orientação inicial do robô (figura 6.3a), a velocidade da bola nesse referencial, no instante de tempo k , é dada por:

$$\begin{bmatrix} v_{x_b}^w(k) \\ v_{y_b}^w(k) \end{bmatrix} = \begin{bmatrix} v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} + \begin{bmatrix} v(k) \\ v_n(k) \end{bmatrix} + d(k)\omega(k) \begin{bmatrix} -\sin(\alpha(k)) \\ \cos(\alpha(k)) \end{bmatrix} \quad (6.21)$$

Definam-se as componentes x e y da posição e velocidade da bola no referencial local do robô, respectivamente $x_b^R, y_b^R, v_{x_b}^R, v_{y_b}^R$. Definam-se ainda $x_b^w, y_b^w, v_{x_b}^w, v_{y_b}^w$ como as mesmas grandezas, mas no referencial global (mundo).

Considere-se agora que o referencial do robô faz um ângulo de θ em relação ao referencial de inércia/global (figura 6.3b). Sabendo que, entre dois instantes de tempo consecutivos, o referencial local do robô rodou $\Delta\theta = \omega\Delta T$, a velocidade da bola neste referencial nos instantes k e $k-1$ é dada respectivamente pelas equações (6.22) e (6.23).

$$\begin{bmatrix} v_{x_b}^w(k) \\ v_{y_b}^w(k) \end{bmatrix} = R_{z,\theta(k)} \left(\begin{bmatrix} v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} + \begin{bmatrix} v(k) \\ v_n(k) \end{bmatrix} + d(k)\omega(k) \begin{bmatrix} -\sin(\alpha(k)) \\ \cos(\alpha(k)) \end{bmatrix} \right) \quad (6.22)$$

$$\begin{bmatrix} v_{x_b}^w(k-1) \\ v_{y_b}^w(k-1) \end{bmatrix} = R_{z,\theta(k-1)} \left(\begin{bmatrix} v_{x_b}^R(k-1) \\ v_{y_b}^R(k-1) \end{bmatrix} + \begin{bmatrix} v(k-1) \\ v_n(k-1) \end{bmatrix} + d(k-1)\omega(k-1) \begin{bmatrix} -\sin(\alpha(k-1)) \\ \cos(\alpha(k-1)) \end{bmatrix} \right) \quad (6.23)$$

onde $R_{z,\theta(k)}$ representa a matriz de rotação de um ângulo θ , em torno do eixo z , tal que:

$$R_{z,\theta(k)} = \begin{bmatrix} \cos(\theta(k)) & -\sin(\theta(k)) \\ \sin(\theta(k)) & \cos(\theta(k)) \end{bmatrix} \quad (6.24)$$

Substituindo (6.22) e (6.23) em (6.17) obtém-se:

$$\begin{aligned}
 R_{z,\theta(k)} \left(\begin{bmatrix} v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} + \begin{bmatrix} v(k) \\ v_n(k) \end{bmatrix} + d(k)\omega(k) \begin{bmatrix} -\sin(\alpha(k)) \\ \cos(\alpha(k)) \end{bmatrix} \right) = \\
 = CR_{z,\theta(k-1)} \left(\begin{bmatrix} v_{x_b}^R(k-1) \\ v_{y_b}^R(k-1) \end{bmatrix} + \begin{bmatrix} v(k-1) \\ v_n(k-1) \end{bmatrix} + d(k-1)\omega(k-1) \begin{bmatrix} -\sin(\alpha(k-1)) \\ \cos(\alpha(k-1)) \end{bmatrix} \right) \Leftrightarrow \\
 \begin{bmatrix} v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} + \begin{bmatrix} v(k) \\ v_n(k) \end{bmatrix} + d(k)\omega(k) \begin{bmatrix} -\sin(\alpha(k)) \\ \cos(\alpha(k)) \end{bmatrix} = \\
 = CR_{z,-\theta(k)} R_{z,\theta(k-1)} \left(\begin{bmatrix} v_{x_b}^R(k-1) \\ v_{y_b}^R(k-1) \end{bmatrix} + \begin{bmatrix} v(k-1) \\ v_n(k-1) \end{bmatrix} + d(k-1)\omega(k-1) \begin{bmatrix} -\sin(\alpha(k-1)) \\ \cos(\alpha(k-1)) \end{bmatrix} \right)
 \end{aligned} \tag{6.25}$$

$R_{z,-\theta(k)} R_{z,\theta(k-1)}$ expressa a rotação do robô em torno de z em dois instantes consecutivos, podendo a rotação ser representada pela equação (6.26). Considerando que entre esses dois instantes (ΔT) o robô rodou um ângulo $\Delta\theta(k) = \omega(k)\Delta T$, a velocidade do robô nos referenciais local e global ficam representadas pelas equações (6.27) e (6.22), respetivamente. Note-se que, embora na equação (6.22) seja necessário conhecer a orientação θ do robô, a equação (6.27) é completamente independente da localização do robô.

$$R_{z,-\theta(k)} R_{z,\theta(k-1)} = R_{z,-\theta(k)+\theta(k-1)} = R_{z,-\omega(k)\Delta T} \tag{6.26}$$

$$\begin{aligned}
 \begin{bmatrix} v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} = - \begin{bmatrix} v(k) \\ v_n(k) \end{bmatrix} - d(k)\omega(k) \begin{bmatrix} -\sin(\alpha(k)) \\ \cos(\alpha(k)) \end{bmatrix} + \\
 + CR_{z,-\omega(k)\Delta T} \left(\begin{bmatrix} v_{x_b}^R(k-1) \\ v_{y_b}^R(k-1) \end{bmatrix} + \begin{bmatrix} v(k-1) \\ v_n(k-1) \end{bmatrix} + d(k-1)\omega(k-1) \begin{bmatrix} -\sin(\alpha(k-1)) \\ \cos(\alpha(k-1)) \end{bmatrix} \right)
 \end{aligned} \tag{6.27}$$

onde

$$R_{z,-\omega(k)\Delta T} = \begin{bmatrix} \cos(\omega(k)\Delta T) & \sin(\omega(k)\Delta T) \\ -\sin(\omega(k)\Delta T) & \cos(\omega(k)\Delta T) \end{bmatrix} \tag{6.28}$$

Definindo x_r e y_r como as coordenadas da origem do referencial local do robô no referencial global, segundo x e y , as posições da bola nos referenciais local e global em função das posições e velocidades anteriores, são dadas respetivamente por:

$$\begin{bmatrix} x_b^R(k) \\ y_b^R(k) \end{bmatrix} = \begin{bmatrix} x_b^R(k-1) \\ y_b^R(k-1) \end{bmatrix} + \begin{bmatrix} v_{x_b}^R(k-1) \\ v_{y_b}^R(k-1) \end{bmatrix} \Delta T \tag{6.29}$$

$$\begin{bmatrix} x_b^w(k) \\ y_b^w(k) \end{bmatrix} = R_{z,\theta(k)} \begin{bmatrix} x_b^R(k) \\ y_b^R(k) \end{bmatrix} + \begin{bmatrix} x_r(k) \\ y_r(k) \end{bmatrix} \tag{6.30}$$

6.4.1 Dimensionamento do Filtro

A presente dissertação propõe o desenvolvimento de um filtro de Kalman que seja capaz de estimar a posição e velocidade de uma bola esférica, a partir de observações num referencial móvel.

O filtro de Kalman consiste essencialmente em duas fases: predição e correção. Na primeira fase, utiliza-se as equações da dinâmica da bola (equações (6.27) e (6.29)) para prever qual será o estado seguinte (6.31), sendo também atualizada a matriz de covariância do erro da estimativa (6.32).

$$\hat{X}^-(k) = A(k)\hat{X}(k-1) + B(k) \quad (6.31)$$

$$P^-(k) = A(k)P(k-1)A^T(k) + Q \quad (6.32)$$

Comece-se por definir o vetor de estado $X(k) \in \mathbb{R}^4$, que contém a posição e velocidade da bola nas componentes x e y no referencial local do robô.

$$X(k) = \begin{bmatrix} x_b^R(k) \\ y_b^R(k) \\ v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} \quad (6.33)$$

Considere-se ainda que $v(k)$, $v_n(k)$ e $\omega(k)$ são entradas cujo valor é conhecido em cada iteração do filtro. Assim, substituindo as equações (6.27) e (6.29) na equação de transição de estado (6.31), obtém-se:

$$\begin{bmatrix} x_b^R(k) \\ y_b^R(k) \\ v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & C \cos(\omega(k)\Delta T) & C \sin(\omega(k)\Delta T) \\ 0 & 0 & -C \sin(\omega(k)\Delta T) & C \cos(\omega(k)\Delta T) \end{bmatrix} \begin{bmatrix} x_b^R(k-1) \\ y_b^R(k-1) \\ v_{x_b}^R(k-1) \\ v_{y_b}^R(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_1(k) \\ b_2(k) \end{bmatrix} \quad (6.34)$$

onde

$$\begin{aligned} b_1(k) &= -v(k) + d(k)\omega(k)\sin(\alpha(k)) + C(\cos(\omega(k)\Delta T)b_{11}(k) + \sin(\omega(k)\Delta T)b_{12}(k)) \\ b_2(k) &= -v_n(k) - d(k)\omega(k)\cos(\alpha(k)) + C(-\sin(\omega(k)\Delta T)b_{11}(k) + \cos(\omega(k)\Delta T)b_{12}(k)) \end{aligned} \quad (6.35)$$

e

$$\begin{aligned} b_{11}(k) &= v(k-1) - d(k-1)\omega(k-1)\sin(\alpha(k-1)) \\ b_{12}(k) &= v_n(k-1) + d(k-1)\omega(k-1)\cos(\alpha(k-1)) \end{aligned} \quad (6.36)$$

Desta forma, a matriz de transição de estado fica:

$$A(k) = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & C \cos(\omega(k)\Delta T) & C \sin(\omega(k)\Delta T) \\ 0 & 0 & -C \sin(\omega(k)\Delta T) & C \cos(\omega(k)\Delta T) \end{bmatrix} \quad (6.37)$$

Analisando a equação de transição de estado (6.34), é possível verificar que o sistema é linear, embora variante no tempo.

Na segunda fase, atualiza-se o ganho do filtro (6.38), utilizam-se as medições da posição da bola oriundas do *software* de visão para corrigir o estado (6.39), e volta-se a atualizar a matriz da covariância do erro (6.40). O ganho vai sendo atualizado cada vez que é realizada uma observação, dando mais ou menos importância às medidas conforme o ganho é maior ou menor, respetivamente.

$$K(k) = P^- H^T (H P^- (k) H^T + R(k))^{-1} \quad (6.38)$$

$$\hat{X}(k) = \hat{X}^-(k) + K(k)(Z(k) - H\hat{X}^-(k)) \quad (6.39)$$

$$P(k) = (I_4 - K(k)H)P^-(k) \quad (6.40)$$

Os robôs da equipa 5DPO apenas conseguem fazer observações da posição da bola através do seu sistema de visão, não conseguindo realizar nenhuma medição de velocidade. Desta forma, a matriz $Z \in \mathbb{R}^2$ é dada por:

$$Z(k) = \begin{bmatrix} x_{bm}^R(k) \\ y_{bm}^R(k) \end{bmatrix} \quad (6.41)$$

Consequentemente, a matriz $H \in \mathbb{R}^{2 \times 4}$ é dada por (6.42) e a matriz do ganho do filtro de Kalman K fica definida em $\mathbb{R}^{4 \times 2}$. Por sua vez, a matriz da covariância da estimativa de estado $P \in \mathbb{R}^{4 \times 4}$.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.42)$$

O ruído das observações foi obtido através de medições da bola realizadas a diferentes distâncias de observação, e com robôs reais. Os resultados dos ensaios serão apresentados na secção 6.4.2. Devido à influência da distância do robô à bola na incerteza da observação, adotou-se um modelo de ruído variável, que será explicado na secção 6.4.3. Desta forma, definindo $\sigma_{rx}^2(k)$, $\sigma_{ry}^2(k)$, $\sigma_{rx}^2(k)$ e $\sigma_{ry}^2(k)$ como sendo as covariâncias das medições da bola no referencial local,

no instante k , segundo as direções x e y , a matriz do ruído de medição $R \in \mathbb{R}^{2 \times 2}$ fica definida por:

$$R(k) = \begin{bmatrix} \sigma_{rxx}^2(k) & \sigma_{rxy}^2(k) \\ \sigma_{ryx}^2(k) & \sigma_{ryy}^2(k) \end{bmatrix} \quad (6.43)$$

No que concerne ao ruído do modelo de transição de estado, a sua obtenção já não é tão trivial. Isto deve-se ao facto de a bola estar num ambiente bastante dinâmico, onde o movimento livre da bola pode ser interrompido a qualquer momento por uma colisão com um ou mais robôs, e cuja intensidade do choque é também desconhecida. Consequentemente, o erro associado à dinâmica da bola fica difícil de prever, sendo impossível analisar com rigor as forças/perturbações que atuam na mesma. Como tal, optou-se por utilizar esta matriz como parâmetro de afinação do filtro, realizando-se ensaios com diferentes valores dentro de ordens de grandeza próximas à do ruído de medição. Estes ensaios serão apresentados e analisados na secção 6.4.5. Considerando σ_{qx}^2 , σ_{qy}^2 , $\sigma_{qv_x}^2$ e $\sigma_{qv_y}^2$ como as variâncias do ruído de processo em x , y , v_x e v_y , a matriz $Q \in \mathbb{R}^{4 \times 4}$ fica definida por:

$$Q = \begin{bmatrix} \sigma_{qx}^2 & 0 & 0 & 0 \\ 0 & \sigma_{qy}^2 & 0 & 0 \\ 0 & 0 & \sigma_{qv_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{qv_y}^2 \end{bmatrix} \quad (6.44)$$

6.4.2 Análise do Ruído de Medição da Bola

A obtenção da observação da bola por parte do robô está sujeita a ruído. Comece por se definir um referencial $x^{R'} Oy^{R'}$ centrado no robô e orientado na direção da bola (figura 6.4). Considere-se ainda o desvio padrão das observações em relação à distância de observação d e a sua direção normal serão denominados por σ_d , σ_{dn} .

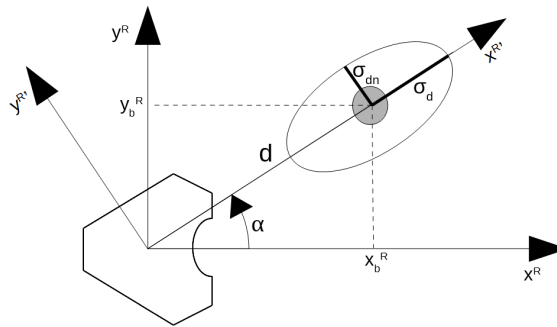


Figura 6.4: Sistema de coordenadas de observação da bola

De modo a determinar esse ruído, realizou-se um teste que consistia na medição da posição da bola a várias distâncias de um robô parado, calculando-se o desvio padrão segundo as direções d e dn para cada caso. Por forma a se obter maior rigor, repetiu-se o teste em três robôs distintos. Os resultados estão visíveis na tabela 6.1.

Tabela 6.1: Desvio padrão das medições da bola

$d(m)$	Desvio padrão σ_r do robô r					
	$\sigma_{d1}(m)$	$\sigma_{dn1}(m)$	$\sigma_{d2}(m)$	$\sigma_{dn2}(m)$	$\sigma_{d3}(m)$	$\sigma_{dn3}(m)$
0.5	7.84×10^{-4}	7.16×10^{-4}	1.31×10^{-3}	3.57×10^{-4}	9.75×10^{-4}	3.76×10^{-4}
1.0	2.02×10^{-3}	7.85×10^{-4}	1.12×10^{-3}	3.58×10^{-4}	9.66×10^{-4}	3.23×10^{-4}
1.5	2.76×10^{-3}	8.33×10^{-4}	2.64×10^{-3}	4.26×10^{-4}	2.16×10^{-3}	5.79×10^{-4}
2.0	4.06×10^{-3}	1.44×10^{-3}	3.30×10^{-3}	5.69×10^{-4}	2.82×10^{-3}	8.03×10^{-4}
2.5	6.05×10^{-3}	1.46×10^{-3}	5.11×10^{-3}	6.70×10^{-4}	3.21×10^{-3}	7.45×10^{-4}
3.0	7.44×10^{-3}	1.40×10^{-3}	6.23×10^{-3}	1.03×10^{-3}	5.50×10^{-3}	9.96×10^{-4}
3.5	1.20×10^{-2}	2.19×10^{-3}	9.98×10^{-3}	6.26×10^{-4}	7.52×10^{-3}	7.52×10^{-4}
4.0	1.51×10^{-2}	1.75×10^{-3}	1.18×10^{-2}	1.50×10^{-3}	9.94×10^{-3}	5.54×10^{-4}
4.5	2.17×10^{-2}	2.07×10^{-3}	1.30×10^{-2}	1.09×10^{-3}	1.32×10^{-2}	9.82×10^{-4}
5.0	1.78×10^{-2}	2.08×10^{-3}	1.53×10^{-2}	1.52×10^{-3}	1.42×10^{-2}	8.01×10^{-4}

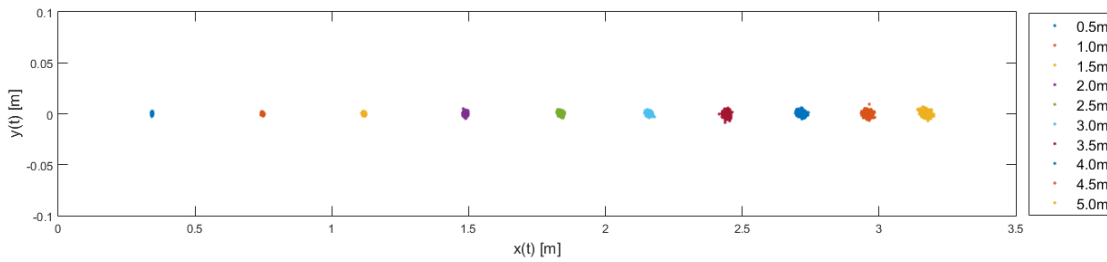


Figura 6.5: Medidas de observação da bola por parte do robô 1

Analisando a tabela 6.1, pode-se concluir que os valores do desvio padrão das medidas foram semelhantes para a mesma distância de observação, e também que este aumenta com a distância. Da mesma forma, pode-se verificar que o desvio padrão das medidas segundo a direção d é maioritariamente superior ao desvio padrão da sua direção normal. Note-se ainda que as pequenas diferenças podem ser explicadas, no sistema de visão, por eventuais desalinhamentos da câmara com o espelho, ou por diferenças na iluminação e calibração das cores (mudando o tamanho e, consecutivamente, o centro do *blob* da bola).

Na sequência deste trabalho, serão considerados os valores de desvio padrão do robô 1, visto que são os valores mais elevados e não se afastam muito dos obtidos para os restantes robôs. Pretende-se, assim, analisar o caso onde se verifica uma observação com maior ruído. A figura 6.5 contém as observações da bola realizadas pelo robô 1.

6.4.3 Ajuste da Covariância do Ruído de Medição da Bola

Analisando os dados obtidos na tabela 6.1 e figura 6.5 pode-se verificar que o desvio padrão das medidas da bola é aproximadamente proporcional à distância a que é realizada a observação.

Tal deve ser tido em conta durante a fase de correção do filtro de Kalman, ou seja, uma observação da bola realizada a uma distância mais próxima do robô deve ter maior importância do que uma observação realizada a uma distância maior. Como tal, utilizou-se uma abordagem semelhante à proposta por Sousa *et al.* [64] para a localização de agentes através de marcos.

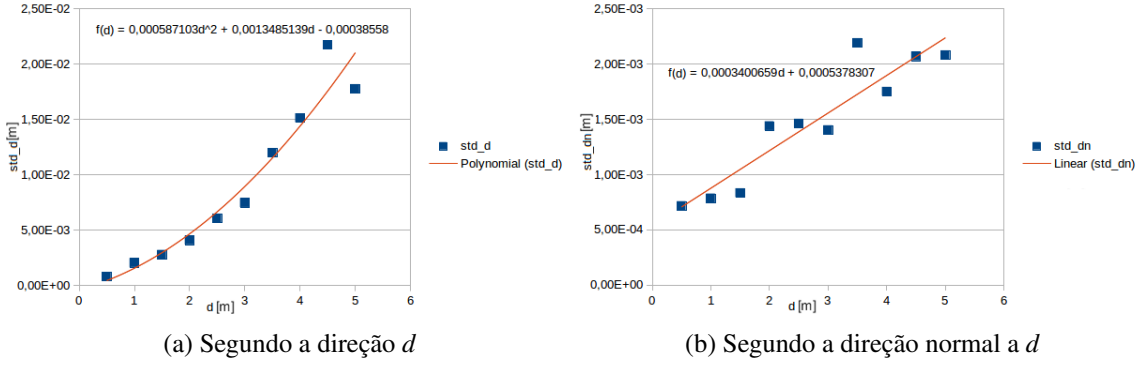


Figura 6.6: Regressão linear e desvio padrão da observação da bola

Assumindo os valores dos desvios padrões segundo a direção d e da sua direção normal do robô 1, tentou-se encontrar as funções que melhor se aproximavam dos pontos medidos (figura 6.6), concluindo-se que os primeiros seriam melhor aproximados por uma função polinomial de segunda ordem (equação (6.45)), e os segundos por uma equação de primeira ordem (equação (6.46)).

$$\sigma_d(d) = 5.871 \times 10^{-4} d^2 + 1.349 \times 10^{-3} d - 3.856 \times 10^{-4} \quad (6.45)$$

$$\sigma_{dn}(d) = 3.401 \times 10^{-4} d + 5.378 \times 10^{-4} \quad (6.46)$$

Como foi referido no final da secção 6.4.1, a matriz de covariância do ruído das medidas do presente filtro de Kalman é uma matriz definida em $\mathbb{R}^{2 \times 2}$, dada por (6.43). Contudo, como se pode ver na mesma equação, a matriz R contém as variâncias segundo as direções x^R e y^R , enquanto que as variâncias apresentadas nas equações 6.45 e 6.46 estão definidas em $x^{R'}$ e $y^{R'}$. Desta forma, é necessário projetar a matriz R' no referencial local do robô, precedendo-se, assim, à rotação da matriz de um ângulo α em torno de z , ou seja, $R_{z,-\alpha}$. A transformação da matriz de covariância entre estes dois referenciais é dada pela equação (6.47) [65]:

$$R = R_{z,-\alpha} R' R_{z,-\alpha}^T \quad (6.47)$$

onde

$$R_{z,-\alpha} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (6.48)$$

$$R' = \begin{bmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_{dn}^2 \end{bmatrix} \quad (6.49)$$

6.4.4 Análise do Ruído de Localização do Robô

Por forma a se obterem resultados mais realistas nos ensaios a apresentar na secção 6.4.5 deve-se ter em conta o ruído associado à localização de cada robô. Como tal, para se fazer uma análise ao mesmo, procedeu-se a um conjunto de medições da localização de três robôs em diferentes pontos do campo. A localização de cada agente é obtida através da fusão sensorial de dados obtidos da odometria, visão e de uma bússola digital. Da informação vinda do *software* de visão, são retirados um conjunto de pontos correspondentes a transições verde/branco sob linhas radiais, estando estes associados às linhas do terreno de jogo. Em diferentes posições do campo, um robô obtém um número diferente de pontos, fazendo com que a incerteza na localização do robô seja afetada pela posição do mesmo em campo. Assim, para ter em conta diversas condições de observação, foram escolhidos quatro posições distintas (figura 6.7).

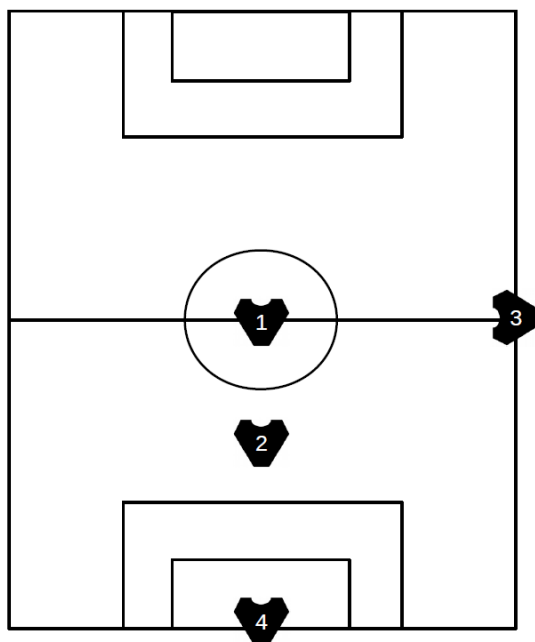


Figura 6.7: Posições de análise de ruído de localização dos robôs

De seguida, procedeu-se ao cálculo do desvio padrão e das variâncias das medidas das diferentes poses dos robôs, estando os resultados presentes nas tabelas 6.2, 6.3 e 6.4.

Tabela 6.2: Desvio padrão e variância das medidas da pose do robô 1 para diferentes pontos

ponto	$\sigma_{x_1}(m)$	$\sigma_{y_1}(m)$	$\sigma_{\theta_1}(rad)$	$\sigma_{x_1}^2(m^2)$	$\sigma_{y_1}^2(m^2)$	$\sigma_{\theta_1}^2(rad^2)$
1	6.41×10^{-4}	8.04×10^{-4}	2.17×10^{-3}	4.12×10^{-7}	6.46×10^{-7}	4.69×10^{-6}
2	1.33×10^{-3}	4.20×10^{-3}	9.07×10^{-4}	1.76×10^{-6}	1.77×10^{-5}	8.22×10^{-7}
3	1.27×10^{-2}	3.92×10^{-3}	3.48×10^{-3}	1.61×10^{-4}	1.54×10^{-5}	1.21×10^{-5}
4	5.25×10^{-4}	2.07×10^{-3}	1.01×10^{-3}	2.76×10^{-7}	4.28×10^{-6}	1.02×10^{-6}

Tabela 6.3: Desvio padrão e variância das medidas da pose do robô 2 para diferentes pontos

ponto	$\sigma_{x_2}(m)$	$\sigma_{y_2}(m)$	$\sigma_{\theta_2}(rad)$	$\sigma_{x_2}^2(m^2)$	$\sigma_{y_2}^2(m^2)$	$\sigma_{\theta_2}^2(rad^2)$
1	1.04×10^{-3}	3.47×10^{-3}	7.39×10^{-1}	1.07×10^{-6}	1.20×10^{-5}	5.46×10^{-5}
2	3.38×10^{-3}	3.06×10^{-3}	1.86×10^{-3}	1.14×10^{-5}	9.37×10^{-6}	3.48×10^{-6}
3	9.94×10^{-2}	7.65×10^{-3}	3.43×10^{-2}	9.89×10^{-3}	5.85×10^{-5}	1.18×10^{-3}
4	1.38×10^{-2}	5.49×10^{-2}	1.30×10^{-1}	1.90×10^{-4}	3.01×10^{-3}	1.69×10^{-4}

Tabela 6.4: Desvio padrão e variância das medidas da pose do robô 3 para diferentes pontos

ponto	$\sigma_{x_3}(m)$	$\sigma_{y_3}(m)$	$\sigma_{\theta_3}(rad)$	$\sigma_{x_3}^2(m^2)$	$\sigma_{y_3}^2(m^2)$	$\sigma_{\theta_3}^2(rad^2)$
1	1.83×10^{-3}	1.96×10^{-3}	2.22×10^{-3}	3.34×10^{-6}	3.85×10^{-6}	4.92×10^{-6}
2	2.54×10^{-3}	1.55×10^{-2}	3.63×10^{-3}	6.48×10^{-6}	2.41×10^{-4}	1.32×10^{-5}
3	1.14×10^{-2}	2.65×10^{-3}	8.20×10^{-4}	1.29×10^{-4}	7.01×10^{-6}	6.73×10^{-7}
4	1.24×10^{-2}	2.60×10^{-2}	6.65×10^{-3}	1.54×10^{-4}	6.77×10^{-4}	4.44×10^{-5}

Para simular o ruído de localização dos robôs nos ensaios das secções 6.4.5.1 e 6.5.2.1, consideraram-se os valores dos desvios padrões obtidos correspondentes à estimativa de variância das medidas da localização obtidas nos ensaios presentes nas tabelas 6.2, 6.3 e 6.4, de acordo com a expressão (6.50) [66].

$$\sigma_x = \sqrt{\frac{\sum_{j=1}^N \sigma_{x,r_j d_i}^2}{N}}, \sigma_y = \sqrt{\frac{\sum_{j=1}^N \sigma_{y,r_j d_i}^2}{N}}, \sigma_\theta = \sqrt{\frac{\sum_{j=1}^N \sigma_{\theta,r_j d_i}^2}{N}} \quad (6.50)$$

onde $\sigma_{x,r_j d_i}^2$ é a variância das observações segundo a direção x do robô j , para a distância i , idem para $\sigma_{y,r_j d_i}^2$ e $\sigma_{\theta,r_j d_i}^2$.

Assim, considera-se que o ruído de localização dos robôs tem um desvio padrão dado por $\sigma_x = 2.96 \times 10^{-2}\text{m}$, $\sigma_y = 1.83 \times 10^{-2}\text{m}$ e $\sigma_\theta = 1.09 \times 10^{-2}\text{rad}$.

No *software* de decisão da equipa, a fusão sensorial da localização nos robôs reais é realizada através de um filtro de Kalman estendido. O estado estimado pelo filtro tem associada uma matriz de covariância, que expressa a confiança no mesmo. Desta forma, na implementação do filtro de Kalman no referencial global que foi realizada no *Decision*, considerou-se a incerteza associada à localização do robô dada por essa matriz de covariância.

6.4.5 Ensaios Realizados

Para validação deste filtro de Kalman e obtenção de resultados para uma futura comparação de abordagens, foram realizados um conjunto de testes a uma implementação deste filtro em Matlab. Os ensaios foram escolhidos de forma a cobrir vários estados do alvo a seguir e do agente que realiza a observação.

O primeiro consistia em ter um robô a descrever uma trajetória de um quadrado de lado 3m, em torno de uma bola parada no centro do campo (figura 6.8a). O robô movimentava-se com velocidade linear de 1m/s e sem velocidade angular, mantendo sempre uma orientação de 0° em relação ao eixo x do referencial global.

O segundo consistia em ter um robô a rodar sobre si mesmo a uma velocidade angular de 0.9rad/s, a uma determinada distância de uma bola estacionária, em relação ao referencial global (figura 6.8b).

O terceiro retratava a situação de um agente parado a observar uma bola em movimento, maioritariamente segundo a direção x do seu referencial, após um remate, sendo que o robô via a bola a afastar-se de si (figura 6.8b).

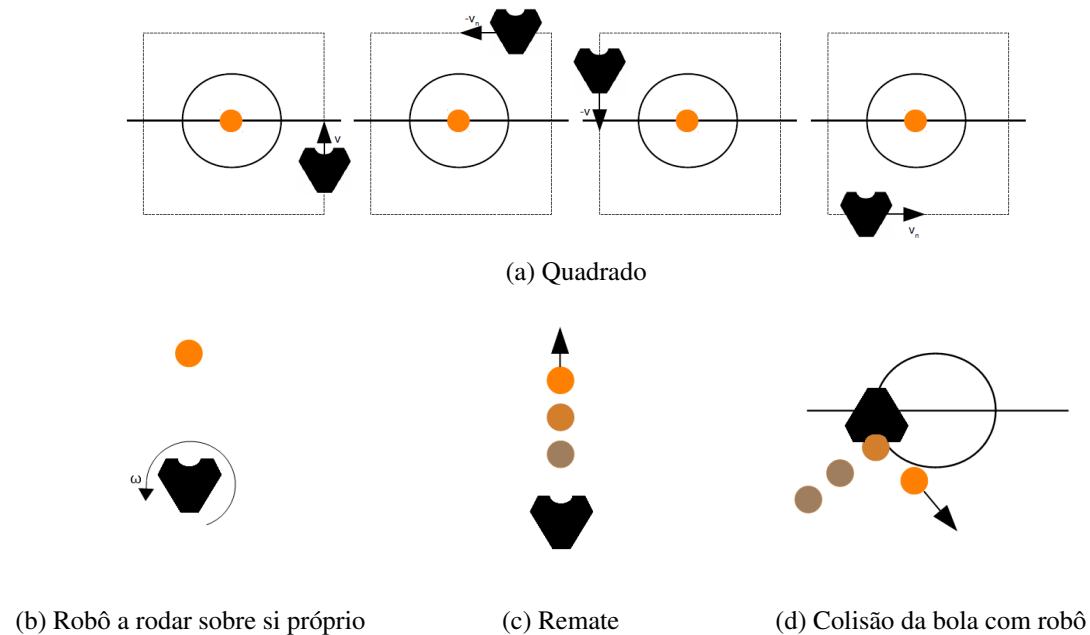


Figura 6.8: Cenários de ensaio do filtro do estimador de posição e velocidade da bola

Desta forma, pretende-se testar a influência das componentes de velocidade do robô perante uma bola estacionária, assim como a resposta do filtro da Kalman a uma bola em movimento com um agente parado.

Foram realizados ensaios com dados obtidos de três formas diferentes, nomeadamente, pose do robô e observação da bola geradas em Matlab, obtidas através de simulações em SimTwo, obtidas a partir de *logs* em ambiente real, num agente da equipa 5DPO. Os dados obtidos em simulação e ambiente real foram guardados num ficheiro, por forma a se conseguirem testar e comparar abordagens diferentes com os mesmos dados de entrada.

Nos ensaios com um robô real, foi realizado um teste adicional que pretendia avaliar situações de mudança de direção da bola. Assim, realizou-se um ensaio em que esta colidia com o agente que estava a realizar a observação, mudando drasticamente de direção (figura 6.8d).

Embora o filtro de Kalman esteja a estimar a posição e velocidade da bola no referencial local do robô, serão também apresentadas a velocidade e, quando for conveniente, a posição da bola no referencial global, determinados a partir das equações (6.22) e (6.30), respetivamente.

6.4.5.1 Ensaios com Dados Gerados no Matlab

Nos ensaios apresentados nas próximas secções, a posição da bola e as posições e velocidades do robô que realiza a observação foram geradas em Matlab.

Para a determinação do valor da matriz de covariância de erro do modelo de transição de estado utilizada neste ensaio, começou-se por definir que as variâncias σ_{qx}^2 , σ_{qy}^2 , $\sigma_{qv_x}^2$ e $\sigma_{qv_y}^2$ seriam iguais à variância de observação de uma bola a 2.5m do robô 1, obtidas através do desvio padrão presente na tabela 6.1, devido a apresentar valores numa ordem de grandeza próxima. De seguida, foram-se ajustando os valores destas covariâncias de modo a obter-se melhores respostas do filtro, tentando filtrar as medidas mas sem tornar a dinâmica do filtro demasiadamente lenta. Assim, reduziu-se o valor das variâncias das posições segundo x e y de modo a dar menos importância às medidas para filtrar o ruído. Quando este valor ficou ajustado, variou-se as variâncias das velocidades de modo a obter o menor erro possível com uma dinâmica aceitável. Nos ensaios que se seguem, apresentar-se-ão os resultados obtidos maioritariamente com a matriz de covariância de erro do modelo de processo Q1 (6.51). Sempre que se achar relevante, mostrar-se-ão resultados com pequenas variações nesta matriz, que serão devidamente identificadas.

$$Q1 = \begin{bmatrix} 3.7 \times 10^{-7} & 0 & 0 & 0 \\ 0 & 3.7 \times 10^{-7} & 0 & 0 \\ 0 & 0 & 3.7 \times 10^{-7} & 0 \\ 0 & 0 & 0 & 3.7 \times 10^{-7} \end{bmatrix} \quad (6.51)$$

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

Nesta secção serão apresentados os ensaios obtidos na situação do robô a descrever uma trajetória de um quadrado em torno de uma bola fixa.

Comece-se por considerar o caso em que o robô está bem localizado, sem nenhum ruído de localização nem de observação da bola. A figura 6.9 pretende ilustrar a situação deste ensaio, contendo a vermelho a trajetória descrita pelo robô e a azul a posição estimada da bola, no referencial global. A figura 6.10a contém as posições medidas e estimadas da bola, no plano xOy do referencial local do robô, enquanto que a figura 6.10b contém o traçado da posição estimada da bola segundo as componentes x e y em função do tempo, no referencial local. Analisando a figura 6.10 com detalhe, é possível verificar que o vetor de estado foi inicializado com uma posição errada, nomeadamente com um metro de desvio nas direções x e y . Esta má inicialização foi propositada, de modo a testar e demonstrar a rapidez de convergência do filtro.

As figuras 6.11a e 6.11b contêm as velocidades estimadas da bola em função do tempo, segundo as componentes x e y do referencial local e global, respetivamente. Como seria de esperar, a velocidade no referencial local apresenta-se como uma onda quadrada com amplitude de 1m/s, correspondendo à velocidade linear do agente em cada aresta do quadrado descrito pela sua trajetória. Convertendo essas velocidades para o referencial global, é possível verificar que esta tende para 0m/s (figura 6.11b). No instante inicial, é visível um pico de 1m/s, que pode ser explicado pelo facto de se ter inicializado a matriz de estado com velocidade nula. Como o estado está referenciado no referencial local, a bola encontra-se parada no referencial global e o robô inicia instantaneamente movimento com velocidade linear igual a 1m/s segundo a direção y , a velocidade da bola no referencial local é igual a -1m/s e não 0m/s, como foi inicializado. Achou-se relevante colocar uma inicialização de velocidade errada para testar a rapidez de convergência do filtro, demonstrando-se que é de facto bastante elevada. Além disso, numa situação real de jogo, o vetor de estado pode ser inicializado com as medidas da bola no instante inicial, mas no que concerne à velocidade, não existe nenhuma estimativa inicial fidedigna.

As pequenas oscilações que são visíveis de 3 em 3 segundos podem ser justificadas por pequenos erros devido à discretização das equações da dinâmica da bola. Recordando a equação utilizada para a predição de velocidade da bola no referencial local (6.47), é visível que a estimação de velocidade depende não só da observação da bola no referencial local, mas também das velocidades lineares e angulares do agente que realiza a observação. Desta forma, uma variação de velocidade do robô em dois instantes de tempo consecutivos resulta num salto em vez de uma transição suave (como aconteceria no caso contínuo), originando, assim, componentes de velocidade em x e y na bola. Relembre-se ainda que, como as posições da bola, agente e velocidade da bola foram geradas em Matlab, e este não está a simular a dinâmica do robô, existe uma variação de velocidade do robô (e consequentemente da observação da bola) maior do que a que existiria num ambiente real. Concluindo, como estas variações são de muito baixa amplitude, podem-se considerar irrelevantes no contexto da aplicação.

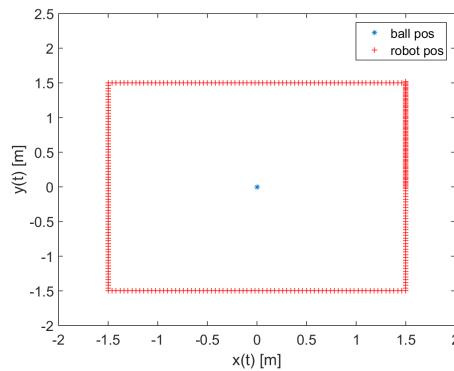
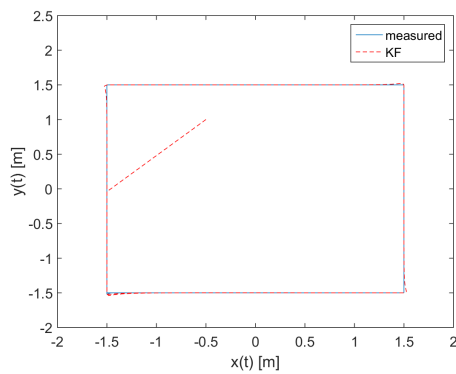
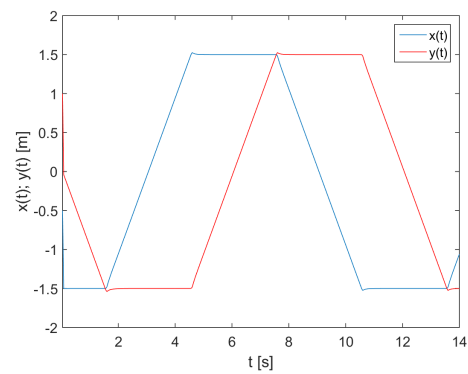


Figura 6.9: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - Posições da bola e do robô, no referencial global

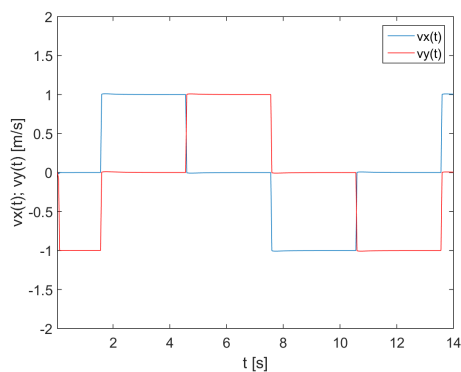


(a) Plano xOy

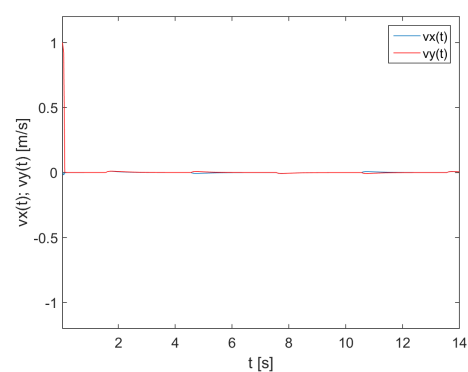


(b) Em função do tempo

Figura 6.10: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - Posição da bola no referencial local



(a) Referencial local



(b) Referencial global

Figura 6.11: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - velocidade da bola

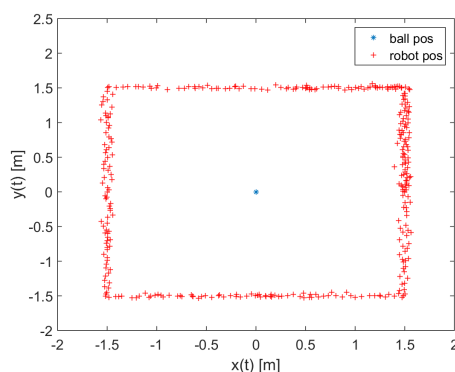
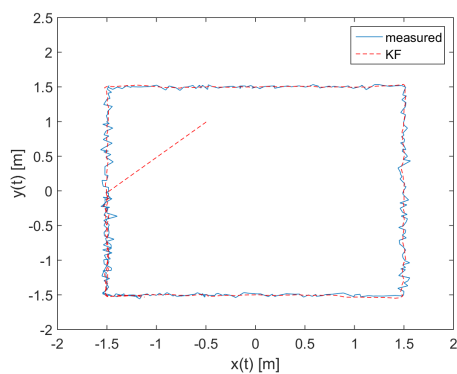
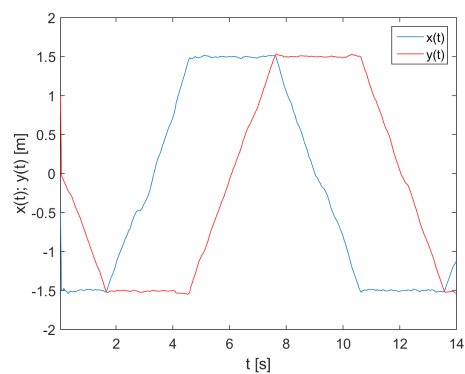


Figura 6.12: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - Posições da bola e do robô, no referencial global

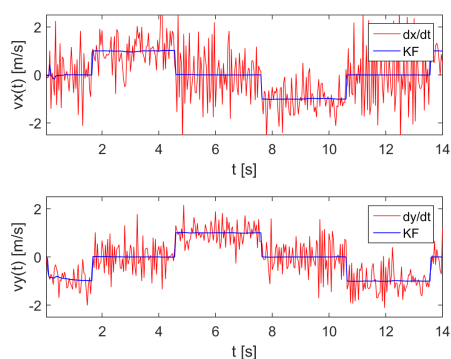


(a) Plano xOy

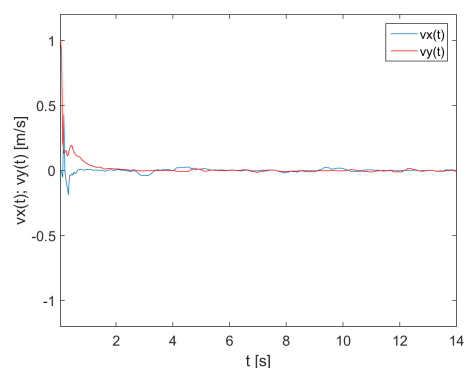


(b) Em função do tempo

Figura 6.13: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, com ruído (Q1) - Posição da bola no referencial local



(a) Referencial local



(b) Referencial global

Figura 6.14: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - velocidade da bola

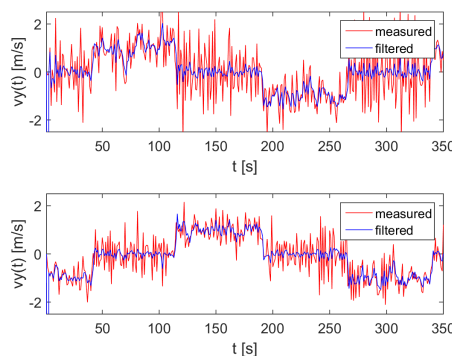


Figura 6.15: Comparação da velocidade estimada através da aproximação numérica da derivada da posição em função do tempo, utilizando as posições medidas ou utilizando as posições estimadas pelo filtro de Kalman

De seguida, adicionou-se ruído Gaussiano de média nula e desvio padrão $\sigma_x = 2.96 \times 10^{-2} \text{m}$, $\sigma_y = 1.83 \times 10^{-2} \text{m}$ e $\sigma_\theta = 1.09 \times 10^{-2} \text{rad}$ à localização do robô, e de acordo com as equações (6.45), (6.46) e (6.47) à observação da bola.

As figuras 6.12, 6.13 e 6.14 apresentam os resultados do ensaio tento em consideração o efeito do ruído, utilizando a matriz de covariância de ruído do modelo de processo Q1. Analisando a figura 6.13a, verifica-se que as observações da posição da bola são devidamente filtradas, e que a estimativa de velocidade também vai de acordo com os valores esperados, tanto no referencial local como no global (figuras 6.14a e 6.14b).

Na figura 6.14a, decidiu-se apresentar uma comparação da velocidade estimada pelo filtro de Kalman, com a estimativa de velocidade a partir da aproximação numérica da derivada da posição em função do tempo. É claramente visível que a estimativa do filtro de Kalman apresenta resultados muito mais suaves e exatos do que a abordagem da derivada. Note-se que, à falta de um sensor externo ou outro método que permita medir a velocidade da bola, assume-se que a velocidade real da bola não estará longe do valor obtido pela aproximação numérica da derivada da posição em função do tempo, à exceção do ruído

A figura 6.15 contém os traçados da velocidade estimada da bola através da aproximação numérica da derivada da posição em função do tempo, utilizando as posições medidas e estimadas pelo filtro. Pode-se verificar que, através da utilização das posições estimadas, se consegue obter traçados com menor ruído do que através da utilização direta das medidas.

Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

Nesta secção serão apresentados os resultados do ensaio em que um robô a rodar sobre si próprio a uma velocidade angular $\omega=0.9\text{rad/s}$, observa uma bola parada, no referencial global, à distância de 1m.

Comece por se considerar o caso em que o robô está bem localizado, sem nenhum ruído de localização nem de observação da bola. A figura 6.16a apresenta as posições medidas e estimadas

da bola, no referencial local do robô. Ao seu lado, a figura 6.16b contém as componentes x e y da posição estimada da bola em função do tempo, também no referencial local. Mais uma vez, o filtro apresentou-se bastante robusto na estimação da posição da bola, resultando numa senoide de amplitude 1m, correspondendo à distância do robô à bola, e frequência de aproximadamente 7 segundos, correspondendo com a velocidade de rotação do robô.

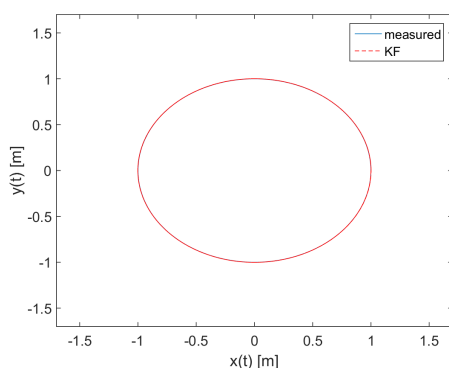
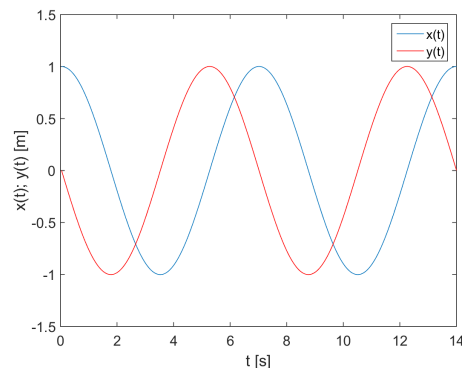
A velocidade da bola no referencial local é representada por uma senoide de amplitude 0.9m/s (figura 6.17a). Tendo em conta que o robô se encontrava à distância de 1m da bola e estava a rodar sobre si mesmo a uma velocidade angular de 0.9rad/s, era de esperar que a sua velocidade tangencial tivesse amplitude de $\omega \times d = 0.9$ m/s. Desta forma, verifica-se que o valor estimado pelo filtro de Kalman vai de acordo ao valor esperado. Após conversão desta velocidade para o referencial global (figura 6.17b), verifica-se que esta tende para 0m/s, como seria de esperar.

Tal como no ensaio anterior, aparece um pico inicial de velocidade no valor de 0.9m/s devido ao facto de o vetor de estado ter sido inicializado com velocidade 0m/s. O estado está referenciado no referencial local, a bola encontra-se parada no referencial global e o robô encontra-se a rodar a uma velocidade angular de 0.9rad/s em relação ao referencial global. Como o robô está a uma distância de 1m da bola, existe um erro de 0.9m/s entre a velocidade que a bola tem no instante inicial e o valor de velocidade com que o estado do filtro foi inicializado.

De seguida, adicionou-se ruído Gaussiano de média nula e desvio padrão $\sigma_x = 2.96 \times 10^{-2}$ m, $\sigma_y = 1.83 \times 10^{-2}$ m e $\sigma_\theta = 1.09 \times 10^{-2}$ rad à localização do robô, e de acordo com as equações (6.45), (6.46) e (6.47) à observação da bola.

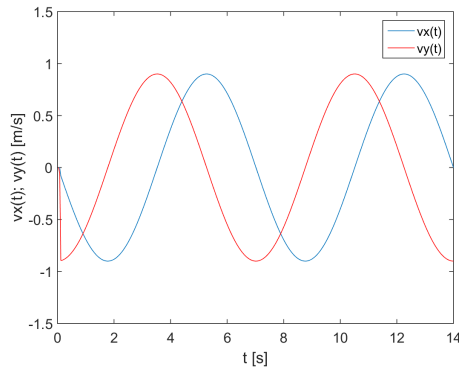
As figuras 6.18 e 6.19 contêm a estimativa de posição e velocidade do filtro de Kalman perante dados com ruído, utilizando a matriz de covariância do ruído do modelo de processo Q1. Analisando estas figuras, é possível verificar que as posições e velocidades estimadas pelo filtro se aproximam dos valores esperados, tal como dos ensaios sem ruído.

Na figura 6.19a é apresentada uma comparação da estimativa de velocidade através do uso do filtro de Kalman e de uma aproximação numérica da derivada da posição em função ao tempo, sendo claramente visível que se obtém melhores resultados com a primeira abordagem.

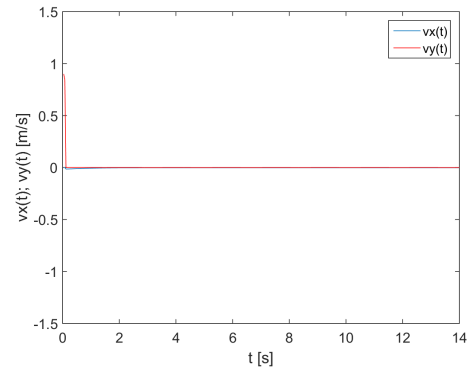
(a) Plano xOy 

(b) Em função do tempo

Figura 6.16: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, sem ruído (Q1) - posição da bola no referencial local

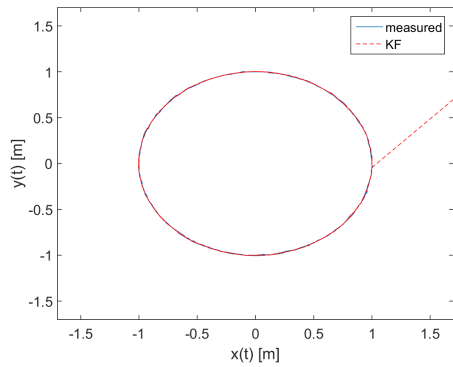
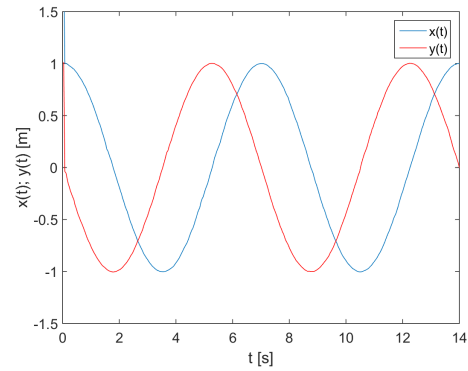


(a) Referencial local



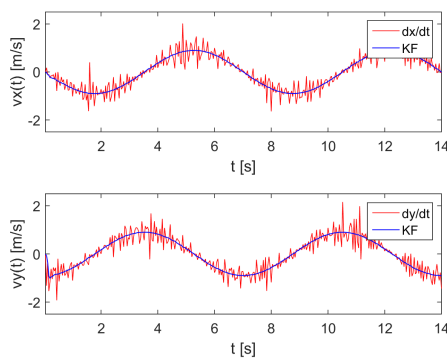
(b) Referencial global

Figura 6.17: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - velocidade da bola

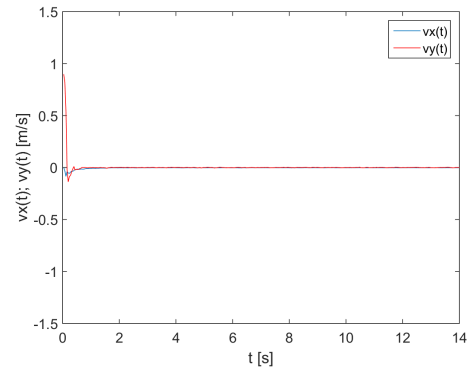
(a) Plano xOy 

(b) Em função do tempo

Figura 6.18: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - posição da bola no referencial local



(a) Referencial local



(b) Referencial global

Figura 6.19: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, com ruído (Q1) - velocidade da bola

Ensaio 3 - Robô Parado e Bola em Movimento

O ensaio apresentado nesta secção consiste num robô parado a observar uma bola a afastar-se de si, segundo a direcção x , simulando uma situação de remate. A bola apresenta uma velocidade inicial de 3m/s e tem um decaimento exponencial de velocidade segundo o modelo da equação (6.5), com o valor de C determinado em (6.20).

Comece por se considerar que o robô está bem localizado, sem nenhum ruído de localização nem na observação da bola. A figura 6.20a contém as observações da bola no plano horizontal, assim como as posições estimadas pelo filtro de Kalman, no referencial local. A forma deste traçado justifica-se pelo facto de o movimento da bola ter sido simulado sem ruído, e de a bola se movimentar unicamente segundo a direcção x . A figura 6.20b apresenta as componentes x e y da posição estimada da bola, em função do tempo, no referencial local.

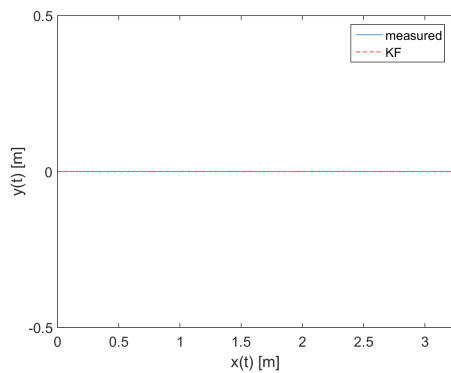
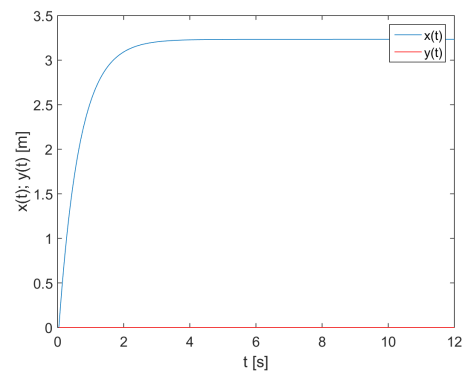
A figura 6.21a contém as componentes x e y da velocidade da bola em função do tempo, no referencial local. Comparando a velocidade estimada pelo filtro com a trajetória descrita pela bola (figura 6.20b), pode-se concluir que a forma de onda da velocidade vai de encontro à trajetória do robô. Relembre-se que a dinâmica da bola seguia um modelo em que a sua velocidade decaía exponencialmente ao longo do tempo.

Olhando para a equação do modelo de transição de estado (6.34), verifica-se que quando o robô não tem velocidade angular (i.e., $\omega=0$), a observação da bola não é tida em conta na estimação da velocidade. Quando o robô está parado (i.e., $v=v_n=\omega=0$), a velocidade apenas é estimada a partir do decaimento exponencial de velocidade face à velocidade no instante anterior. Como o vetor de estado da bola é inicializado com velocidade 0m/s, e a velocidade apenas é estimada pelo decaimento em relação ao instante anterior, a velocidade nunca iria ser diferente de 0m/s. Contudo, e como se pode verificar na figura 6.21a, o filtro de Kalman conseguiu estimar a velocidade da bola a partir das observações realizadas pelo robô durante a fase de correção.

A velocidade da bola no referencial global coincide com a velocidade no referencial local, devido ao facto de o robô se encontrar parado, com uma orientação de 0° em relação ao referencial global (figura 6.21b).

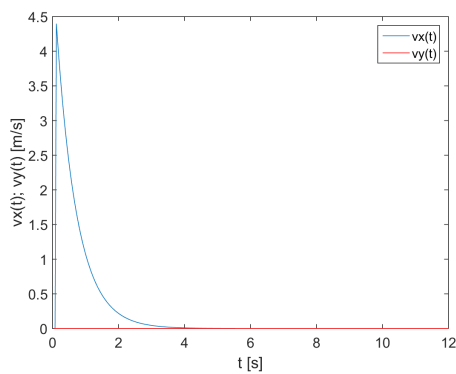
De seguida, adicionou-se ruído Gaussiano de média nula e desvio padrão $\sigma_x = 2.96 \times 10^{-2}\text{m}$, $\sigma_y = 1.83 \times 10^{-2}\text{m}$ e $\sigma_\theta = 1.09 \times 10^{-2}\text{rad}$ à localização do robô, e de acordo com as equações (6.45), (6.46) e (6.47) à observação da bola.

As figuras 6.22 e 6.23 contêm a posição e velocidade estimadas pelo filtro de Kalman com a matriz de covariância do modelo de transição de estado Q1, considerando o ruído de observação da bola e localização do robô anteriormente descrito. Analisando as figuras, é possível verificar que as posições e velocidades estimadas pelo filtro se aproximam dos resultados obtidos sem ruído e, à semelhança destes, próximos dos valores esperados.

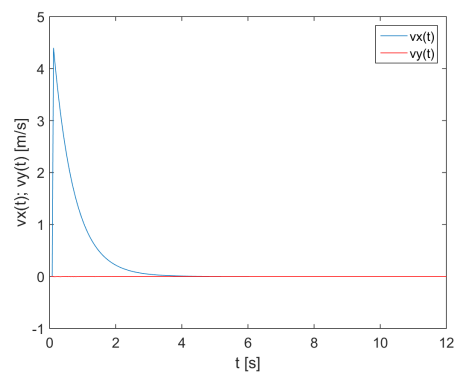
(a) Plano xOy 

(b) Em função do tempo

Figura 6.20: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, sem ruído (Q1) - posição da bola no referencial local

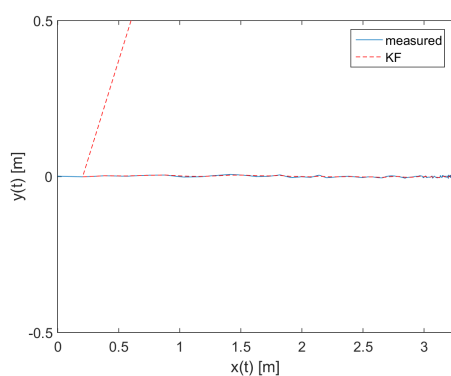
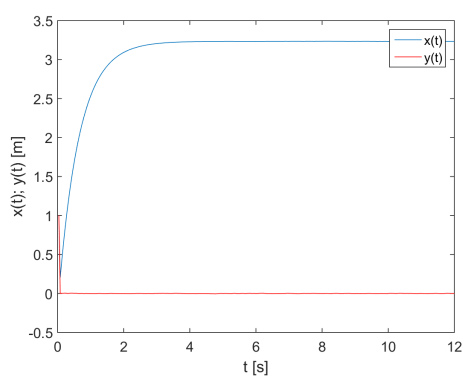


(a) Referencial local



(b) Referencial global

Figura 6.21: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados gerados em Matlab, sem ruído (Q1) - velocidade da bola

(a) Plano xOy 

(b) Em função do tempo

Figura 6.22: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - posição da bola no referencial local

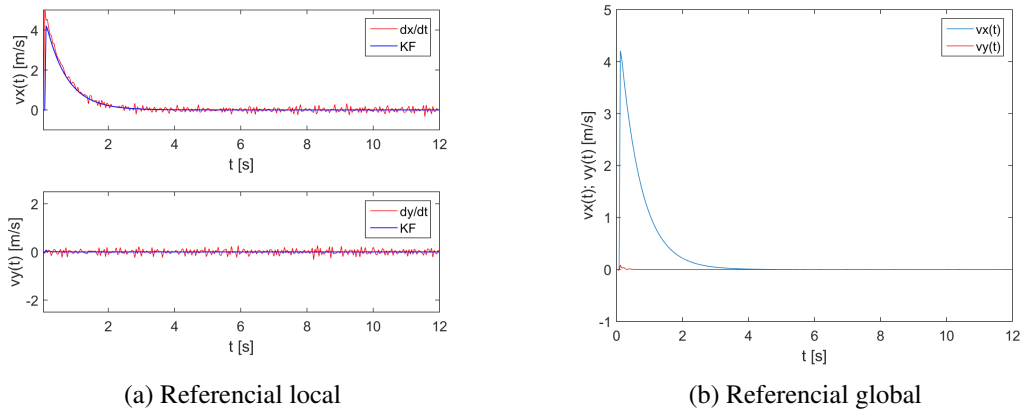


Figura 6.23: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados gerados em Matlab, com ruído (Q1) - velocidade da bola

A figura 6.23a contém uma comparação da velocidade estimada da bola pelo filtro de Kalman e pela aproximação numérica da derivada da posição em função do tempo. Embora seja visível que, durante o instante inicial, a velocidade estimada pelo filtro de Kalman não conseguiu atingir os 5m/s como a velocidade estimada pela derivada, verifica-se que esta tende rapidamente para o valor correto, apresentando uma curva mais suave. Isto apresenta uma grande vantagem, principalmente se a finalidade da estimativa de velocidade for a sua utilização no controlo da velocidade do robô durante uma receção da bola, por exemplo.

6.4.5.2 Ensaios com Dados Obtidos no Simulador

Nas próximas secções serão apresentados os ensaios descritos em 6.4.5, em que os dados de entrada do filtro de Kalman foram obtidos através de *logs* retirados no *software* de decisão da equipa 5DPO, recorrendo a cenários simulados em SimTwo. Desta forma, as velocidades e ações do robô são definidas a partir do *Decision*, enquanto que as posições do robô e da bola são obtidas no simulador, possuindo o simulador um motor de física que simula a dinâmica do robô e da bola.

Para tornar a simulação mais realista, foi adicionado, no simulador, ruído à observação da bola por parte dos agentes na mesma ordem de grandeza do ruído existente nos robôs reais. Este ruído tinha em conta a distância a que o robô se encontrava da bola, de acordo com as expressões (6.45), (6.46) e (6.47).

Durante os ensaios foi utilizada a mesma matriz de covariância do modelo de transição de estados utilizada na secção 6.4.5.1, nomeadamente, dada pela matriz (6.51).

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

O ensaio apresentado nesta secção consiste num robô a descrever uma trajetória de um quadrado com lado 3m em torno de uma bola estacionária, a uma velocidade linear de aproximadamente 1m/s, e com uma orientação de 0° em relação ao eixo x do referencial global.

A figura 6.24 contém as posições estimadas da bola e do robô que está a realizar a observação, no referencial global, podendo, assim, visualizar-se a trajetória executada pelo robô. Como se pode verificar, a trajetória é ligeiramente diferente à apresentada nos ensaios em Matlab (figura 6.12), devido ao facto do simulador simular não só a cinemática do robô, como também a sua dinâmica, em conjunto com o controlador de trajetórias. A figura 6.25a apresenta uma comparação das posições observadas e estimadas da bola no plano xOy , no referencial local. A figura 6.25b contém as componentes x e y das posições estimadas da bola em função do tempo, também no referencial local.

A figura 6.26 contém a a velocidade estimada da bola no referencial local (figura 6.26a) e no referencial global (figura 6.26b). Pode-se concluir que este estimador conseguiu estimar com sucesso a velocidade da bola, quer no referencial local, quer no referencial global. Comparando a figura 6.26a com a respetiva figura do ensaio em Matlab (figura 6.12), pode-se verificar que as formas de onda não são exatamente iguais. Isto deve-se principalmente ao facto de o controlador de trajetórias utilizado neste ensaio ser o controlador que é executado nos robôs reais, estando a ser simulado num simulador capaz de simular uma dinâmica bastante aproximada da dinâmica de um robô real, enquanto que no ensaio anterior as trajetórias eram ideais, geradas através de um *script* em Matlab. Neste ensaio utilizou-se um controlador *GoToXYTheta*, as tarefas executadas pelo agente eram do tipo *GoToPoint*, pelo que o robô sofria uma desaceleração quando se aproximava do ponto destino (vértices do quadrado). Não obstante, verifica-se que a velocidade no referencial global tende rapidamente para 0m/s, indo de encontro com a velocidade real da bola.

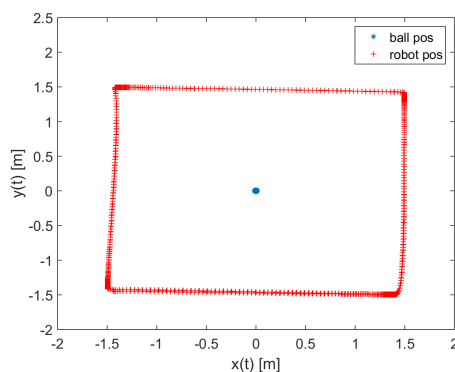


Figura 6.24: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - Posições da bola e do robô, no referencial global

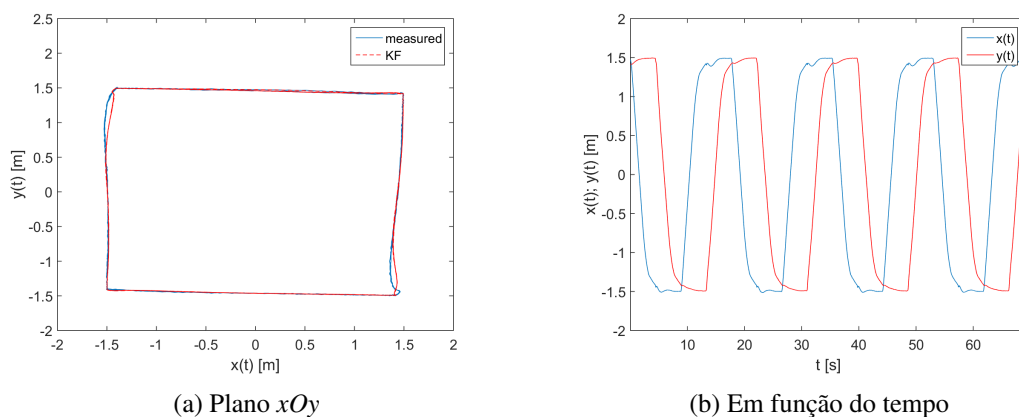


Figura 6.25: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - Posição da bola no referencial local

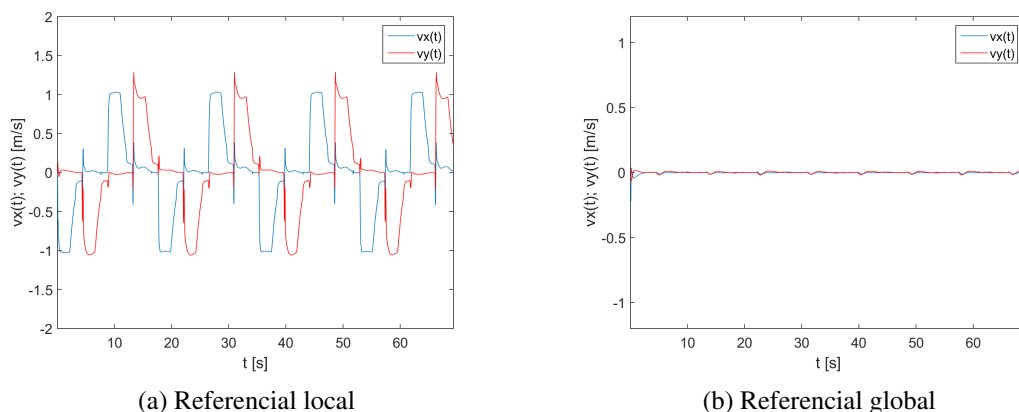


Figura 6.26: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - velocidade da bola

À semelhança do ensaio anterior, nos traçados da velocidade estimada da bola, voltam a ser visíveis uns picos periodicamente. Estes devem-se na mesma a erros devido à discretização das equações da dinâmica da bola. Todavia, verifica-se que estes podem ser desprezáveis, tendo em conta a sua ordem de grandeza.

Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

No ensaio apresentado nesta secção, um robô rodava sobre si próprio a uma velocidade angular de $\omega=0.9\text{rad/s}$, enquanto observava uma bola estacionária, no referencial global, a uma distância de 1m.

A figura 6.27a contém as posições medidas e estimadas da bola no plano xOy , no referencial local do robô. A figura 6.27b apresenta as posições da bola estimadas em função do tempo, no referencial local.

As figuras 6.28a e 6.28b contêm as componentes x e y da velocidade estimada da bola nos referenciais local e global, respectivamente.

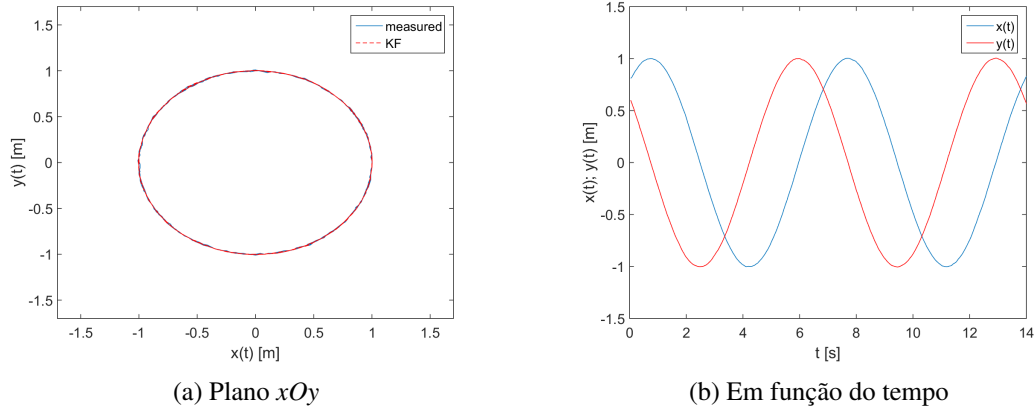


Figura 6.27: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos em SimTwo (Q1) - posição da bola no referencial local

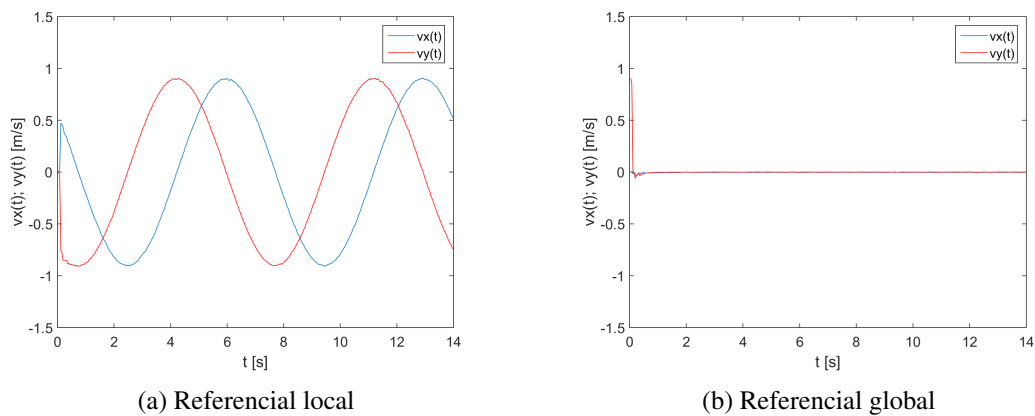


Figura 6.28: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - velocidade da bola

Como se pode verificar, o filtro de Kalman conseguiu estimar a posição e velocidade da bola de acordo com o seu valor esperado. A sua posição é representada por uma senoide com amplitude de 1m, correspondendo à distância do robô à bola, e frequência de aproximadamente 7 segundos, correspondendo com a velocidade de rotação do robô. No que concerne à velocidade estimada na bola, verifica-se que no referencial local é expressa por uma senoide de amplitude 0.9rad/s, correspondendo à velocidade angular a que o robô rodava. Convertendo a velocidade para o referencial global, verifica-se que esta tende rapidamente para zero.

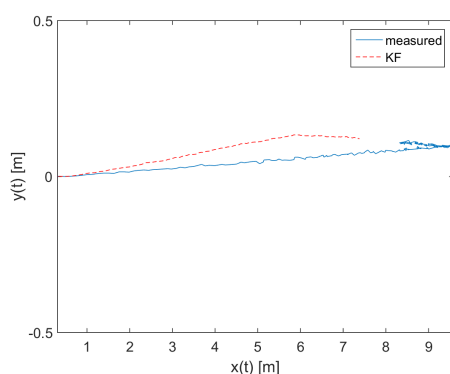
Note-se, ainda, que o pico inicial presente na figura 6.28b pode ser explicado pela inicialização do vetor de estado com velocidades em x e y de 0m/s. O estado está referenciado no referencial local, a bola encontra-se parada no referencial global, e o robô encontra-se a rodar a uma velocidade angular de 0.9rad/s em relação ao referencial global. Como o robô está a uma distância de

1m da bola, existe um erro de 0.9m/s entre a velocidade que a bola tem no instante inicial e o valor de velocidade com que o estado do filtro foi inicializado. No entanto, optou-se por se deixar esta estimativa errada de velocidade inicial para testar e demonstrar a rapidez de convergência deste estimador.

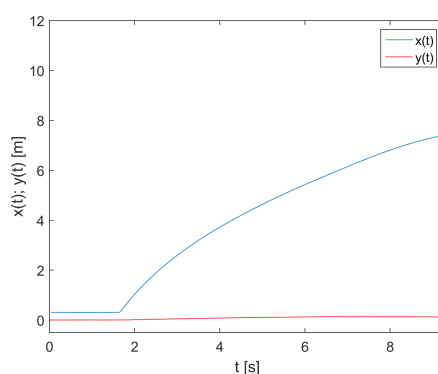
Ensaio 3 - Robô Parado e Bola em Movimento

Nesta secção será apresentado um ensaio em que um robô estacionário observa uma bola em movimento maioritariamente segundo a direção x do referencial global, após um remate rasteiro de baixa intensidade.

A figura 6.29a apresenta as posições observada e estimada da bola no plano xOy , no referencial local. A figura 6.29b contém as componentes x e y da posição estimada da bola, no referencial local.

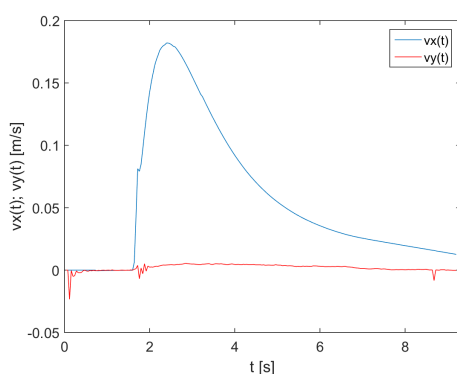


(a) Plano xOy

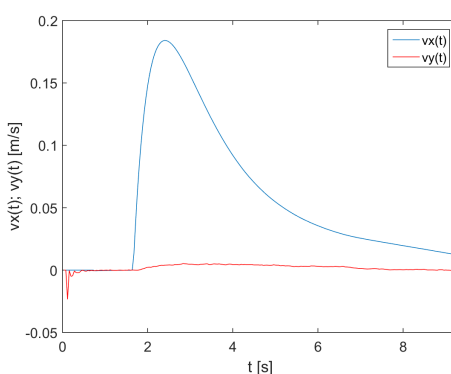


(b) Em função do tempo

Figura 6.29: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos em SimTwo (Q1) - posição da bola no referencial local



(a) Referencial local



(b) Referencial global

Figura 6.30: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q1) - velocidade da bola

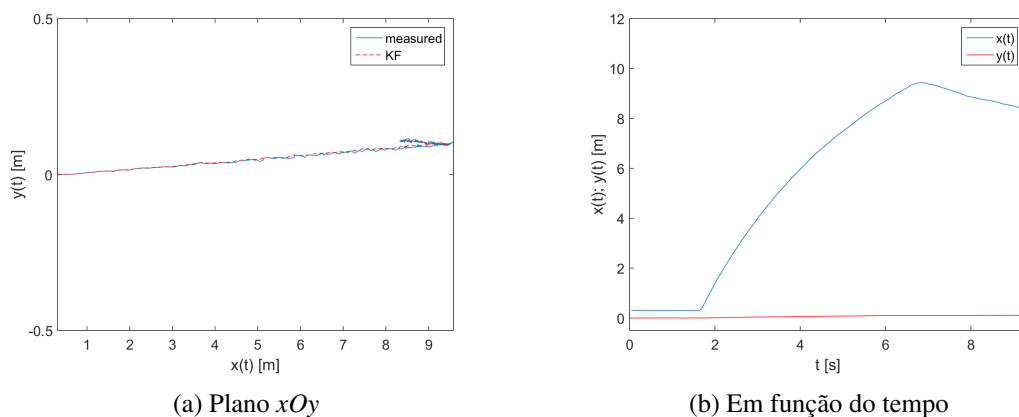


Figura 6.31: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos em SimTwo (Q2) - posição da bola no referencial local

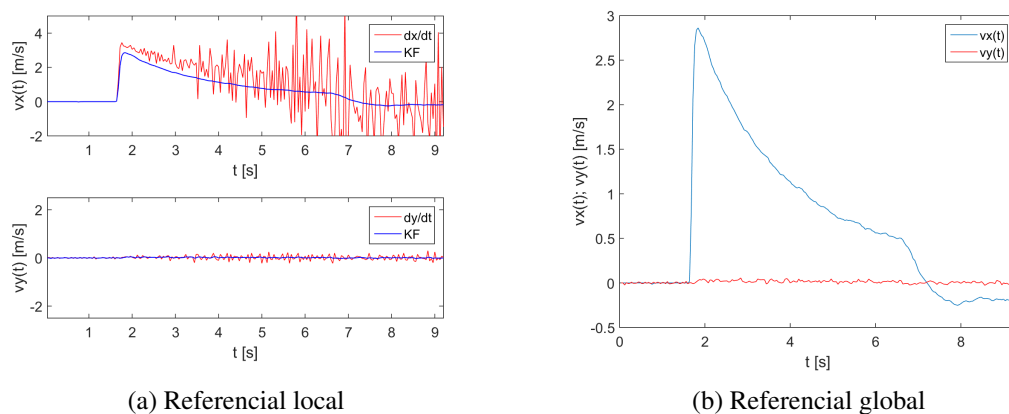


Figura 6.32: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos em SimTwo (Q2) - velocidade da bola

Nas figuras 6.30a e 6.30b são apresentadas as componentes x e y da velocidade estimada da bola, no referencial local e global, respetivamente.

Analisando a figura 6.29a, pode-se verificar que a estimativa de posição da bola do filtro não conseguiu acompanhar a trajetória da mesma. A bola não chegou a atingir a baliza, que estava localizada aproximadamente a uma distância de 9.5m do robô, e a linha de trajetória também não foi a mais adequada. Da mesma forma, observando a figura 6.30, verifica-se que a velocidade da bola também esteve longe do valor real. Isto deve-se essencialmente ao facto de o modelo da dinâmica da velocidade da bola utilizado na fase de transição de estado resultar de uma simplificação do movimento da bola, assim como o facto de o simulador conseguir reproduzir um movimento da bola bastante realista. Como os modelos simulado e utilizado na fase de predição de estado são diferentes, exige um maior esforço do filtro de Kalman na fase de correção para corrigir as diferenças dos dois modelos, através das observações realizadas para a bola. Por este motivo, realizou-se uma nova previsão de posição e velocidade, com os mesmos dados de entrada, mas dando uma

maior importância às observações da bola, através do aumento das variâncias $\sigma_{v_x}^2$ e $\sigma_{v_y}^2$, resultando na matriz de covariância do modelo de transição de estado Q2 (equação (6.52)).

$$Q2 = \begin{bmatrix} 3.7 \times 10^{-7} & 0 & 0 & 0 \\ 0 & 3.7 \times 10^{-7} & 0 & 0 \\ 0 & 0 & 3.7 \times 10^{-4} & 0 \\ 0 & 0 & 0 & 3.7 \times 10^{-4} \end{bmatrix} \quad (6.52)$$

As figuras 6.31 e 6.32 apresentam os resultados do ensaio para a estimação com a nova matriz de covariância Q2. Analisando a figura 6.32, pode-se verificar que dando mais importância às observações da bola, a velocidade estimada pelo filtro de Kalman já fica mais próxima da velocidade real. Comparando com a figura 6.30, pode-se verificar que dando mais importância às medidas da bola resulta em maior ruído na estimativa de velocidade, porém, este apresenta valores perfeitamente aceitáveis no contexto da aplicação. Na figura 6.32a é ainda realizada uma comparação entre a posição estimada pelo filtro de Kalman com a estimada através da derivada da posição em função do tempo. Embora a velocidade estimada pelo filtro esteja ligeiramente abaixo da sua velocidade real, é visível que o estimador conseguiu acompanhar estimar a mesma a menos de um pequeno *offset*, com uma curva bastante mais suave.

Embora não se tenham alterado as variâncias σ_x^2 e σ_y^2 , verifica-se que a posição estimada pelo filtro já se aproxima mais da trajetória descrita pela bola (figura 6.31a). Desta forma, verifica-se que o desvio observado na figura 6.29a se devia à má estimativa de velocidade. Pode-se ainda observar que a posição estimada pela bola foi capaz de acompanhar a mudança de direção da bola quando esta colidiu com a baliza, aproximadamente aos 9.5m (figura 6.31a) e 6.5s (figura 6.31).

Convertendo a velocidade da bola para o referencial global, obtém-se um traçado semelhante ao da velocidade no referencial local, devido ao facto de o robô se encontrar parado e com uma orientação de 0° em relação ao referencial global.

6.4.5.3 Ensaios com Dados Obtidos num Robô Real

Nas próximas secções serão apresentados os ensaios do filtro de Kalman descritos na secção 6.4.5, com dados recolhidos em ambiente real através de *logs* retirados no *software* de decisão, estando este a ser executado num robô da equipa 5DPO. Desta forma, os dados de entrada estão afetados de observação da bola e de localização do robô existente numa situação real.

Visto que no ensaio 6.4.5.2 se verificou que, pelo menos no simulador, era necessário dar mais importância às observações da bola na estimação da sua velocidade, decidiu-se realizar os próximos ensaios utilizando as matrizes de covariância Q1 e Q2, respetivamente dadas por (6.51) e (6.52).

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

Nesta secção será apresentado o ensaio de um robô a descrever a trajetória de um quadrado de lado 3m em torno de uma bola estática, a uma velocidade linear de aproximadamente 1m/s e com uma orientação de 0° em relação ao referencial global. Comece por se considerar que o filtro de Kalman utiliza a matriz de covariância do modelo de transição de estado Q1.

A figura 6.33 contém as posições estimadas da bola, assim como as posições ocupadas pelo robô que realiza a observação, no referencial global. A figura 6.34a apresenta as posições medidas e estimadas da bola, no referencial local. Na figura 6.34b pode-se observar as componentes x e y da posição estimada da bola em função do tempo, no referencial global.

O robô não descreve um quadrado perfeito devido ao facto de as trajetórias serem planeadas pelo controlador de trajetórias *GoToXYTheta*, implementado no *software* de decisão. Em qualquer destas três imagens é visível que, embora a bola tenha permanecido parada em relação ao referencial global, esta aparenta ter-se movido.

As figuras 6.35a e 6.35b apresentam a velocidade estimada da bola em função do tempo, nos referenciais local e global, respetivamente. A velocidade estimada da bola, no referencial local, apresenta a forma de uma onda quadrada com 1m/s de amplitude, devido ao facto de o robô se estar a deslocar maioritariamente com esta velocidade. À semelhança do que aconteceu no ensaio 6.4.5.2, o controlador de trajetórias diminui a velocidade do robô quando este se aproxima dos pontos destino (vértices do quadrado), refletindo-se numa menor velocidade de observação da velocidade da bola nesses instantes. Contudo, é visível que a velocidade estimada apresenta bastante ruído. Este pode ser explicado por erros de discretização das equações da dinâmica da velocidade da bola, mas principalmente devido à influência do ruído presente nas estimativas de velocidade do robô, vindas da odometria, que é injetado diretamente na equação de transição de estado. A forma de onda das velocidades lineares do robô estimadas pela odometria do robô estão representadas na figura 6.38, sendo visível o ruído anormal que se encontra nestes sinais. Convertendo a velocidade para o referencial global, este último efeito é anulado, obtendo-se um sinal mais limpo (figura 6.35).

A ondulação de pequena amplitude que se verifica nesta figura pode-se justificar pelo desvio entre a distância medida pelo agente e a distância real a que esta se encontrava, visto que, na perspetiva do robô, a bola estava-se a movimentar no referencial global (figura 6.33).

A ligeira oscilação presente no instante inicial deve-se ao facto de se ter inicializado o vetor de estado com uma velocidade igual a 0m/s. Como este vetor está referenciado no referencial local, a bola na realidade encontra-se parada no referencial global e no instante inicial o robô está em movimento com uma pequena velocidade, a velocidade da bola no referencial local não é exatamente zero. Contudo, decidiu-se deixar esta inicialização errada para testar e provar a rapidez de convergência do estimador. Este pico é inferior em relação a ensaios anteriores visto que, devido à inércia e dinâmica do robô real, este não acelera instantaneamente, fazendo com que a diferença entre o valor da velocidade da bola no referencial local e o valor com o qual o filtro foi inicializado não seja tão grande.

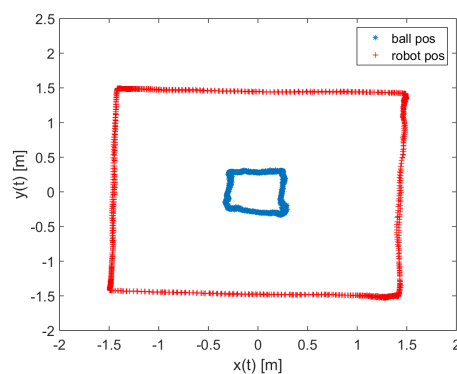
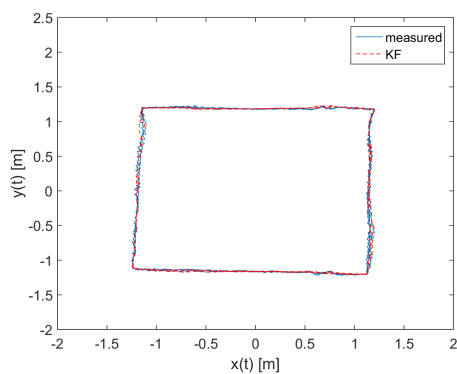
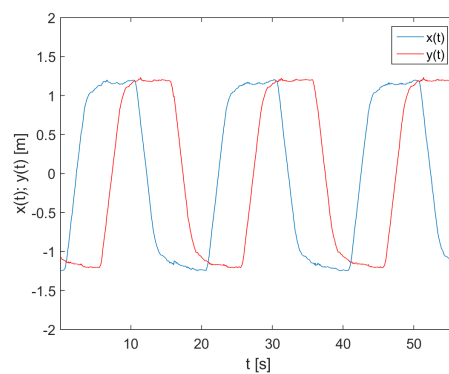


Figura 6.33: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - Posições da bola e do robô, no referencial global

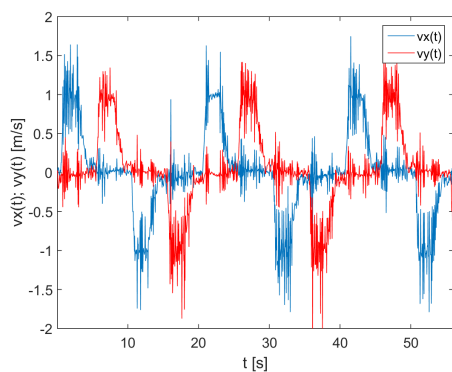


(a) Plano xOy

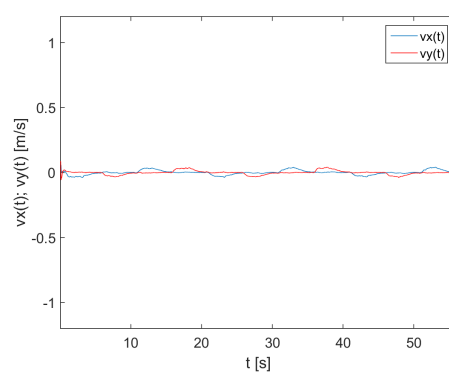


(b) Em função do tempo

Figura 6.34: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - Posição da bola no referencial local



(a) Referencial local



(b) Referencial global

Figura 6.35: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - velocidade da bola

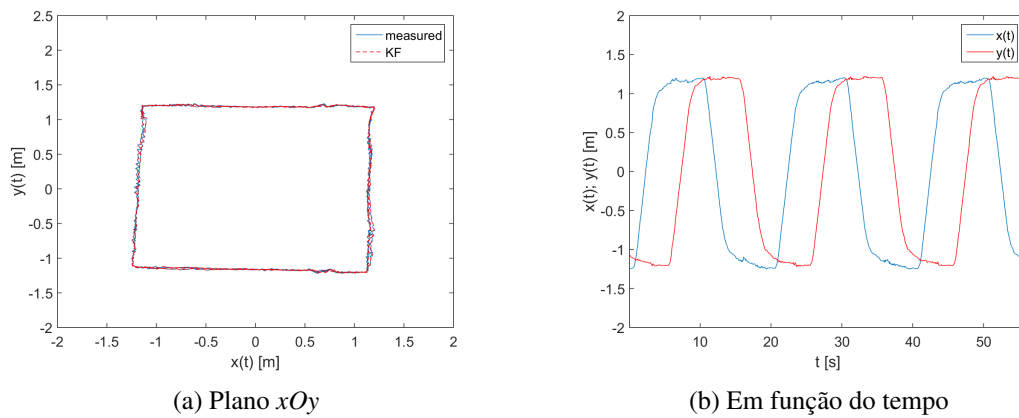


Figura 6.36: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - Posição da bola no referencial local

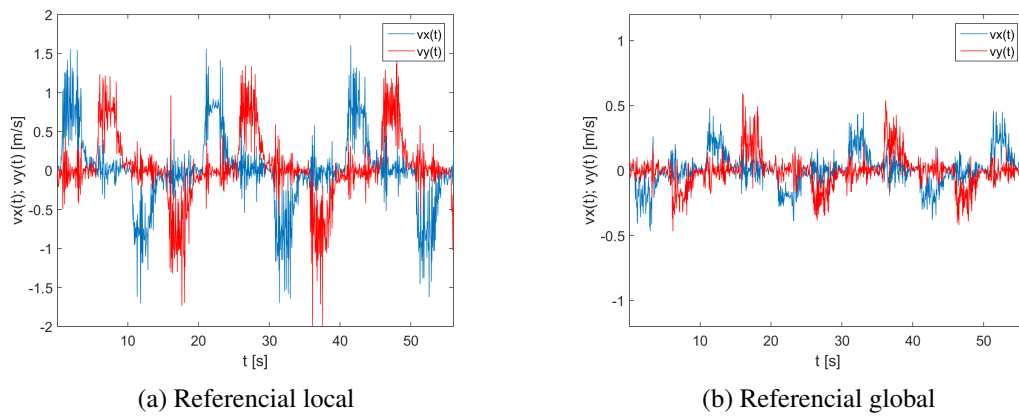


Figura 6.37: Ensaio 1 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola

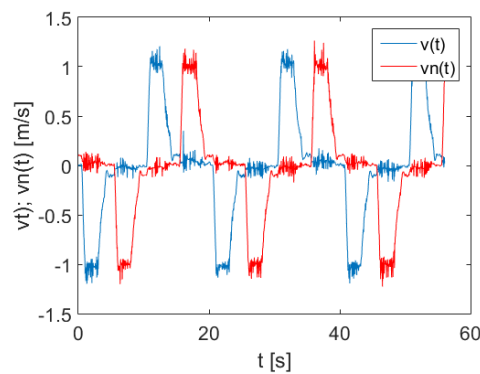


Figura 6.38: Traçado das velocidades lineares do robô

Como no ensaio 6.4.5.2 se verificou que era necessário dar mais importância às medidas da

bola para que este estimador conseguisse acompanhar a dinâmica da mesma, decidiu-se apresentar os resultados obtidos utilizando a matriz de covariância do modelo de transição de estado Q2, para os mesmos dados de entrada.

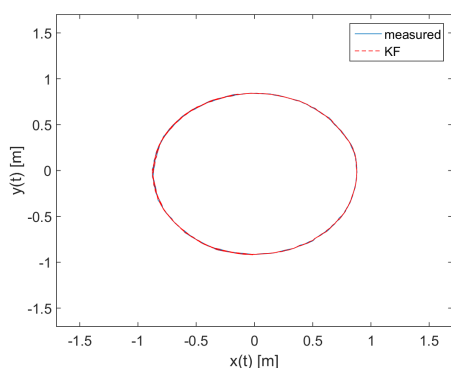
As figuras 6.36 e 6.37 contêm as posições e velocidades estimadas pelo filtro de Kalman com a matriz de covariância Q2.

Como se pode verificar, a estimativa de posição da bola não sofre grandes diferenças, contudo, dando uma maior importância às observações resulta numa estimativa de velocidade com o mesmo valor médio, mas com mais ruído. Isto significa que, embora a matriz de covariância do modelo de transição de estado Q1 resulte em boas estimativas no caso da bola estar parada, não consegue acompanhar com precisão uma bola em movimento. Desta forma, pode-se concluir que, a menos que se saiba que a bola está parada (por exemplo, momentos que sucedam uma *set piece*), não se deve utilizar a matriz de covariância Q1. A determinação da matriz Q é algo não trivial, que deve ser feito com bastante cuidado, sob a pena de não conseguir detetar com rigor movimentos da bola.

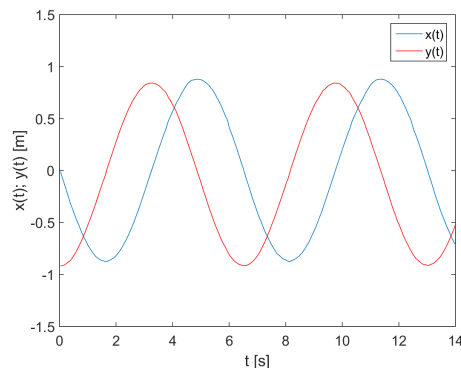
Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

Nesta secção será analisado o ensaio em que um robô roda sobre si próprio a uma velocidade de aproximadamente 0.9rad/s, a uma distância de 1m de uma bola parada em relação ao referencial global.

O gráfico da figura 6.39a contém a posição do robô e as posições estimadas da bola no plano xOy , do referencial local. A figura 6.39b apresenta as componentes x e y das posições estimadas da bola em função do tempo, no referencial global. Embora a senoide apresente um período que vai de acordo com a velocidade de rotação do robô, pode-se verificar que, devido a erros na calibração da câmara, o robô apenas consegue observar a bola a cerca de 0.9m, quando esta está na realidade a 1m do mesmo.

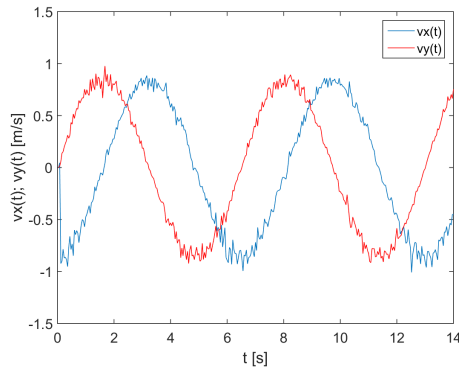


(a) Plano xOy

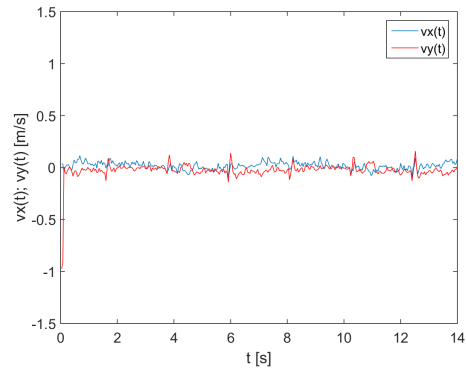


(b) Em função do tempo

Figura 6.39: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=0.9\text{rad/s}$, $d=1\text{m}$)

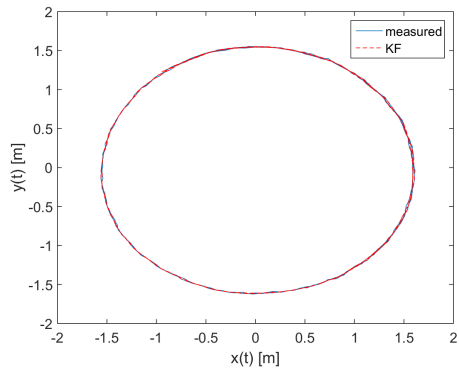
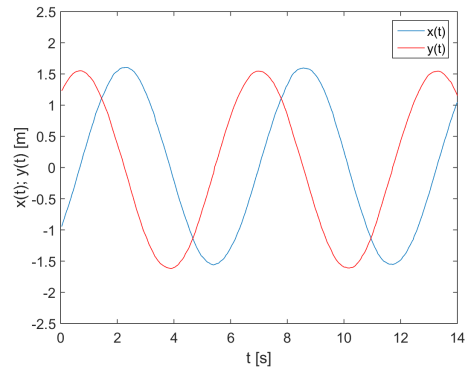


(a) Referencial local



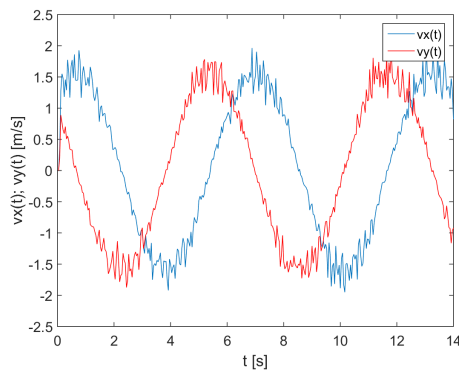
(b) Referencial global

Figura 6.40: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=0.9\text{rad/s}$, $d=1\text{m}$)

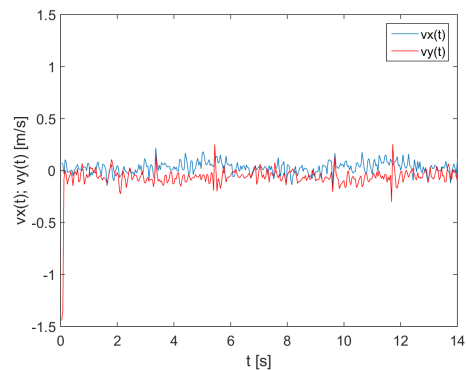
(a) Plano xOy 

(b) Em função do tempo

Figura 6.41: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=0.9\text{rad/s}$, $d=2\text{m}$)

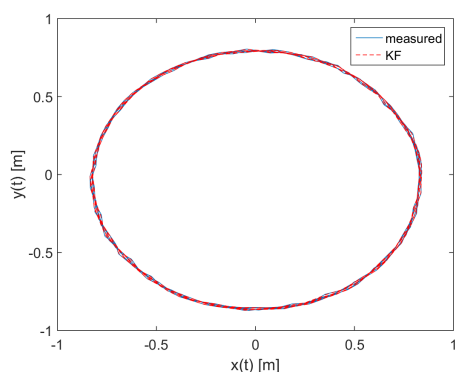
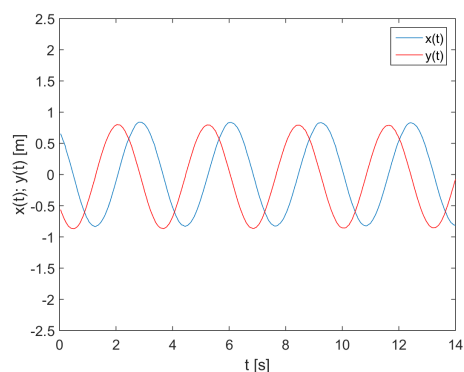


(a) Referencial local



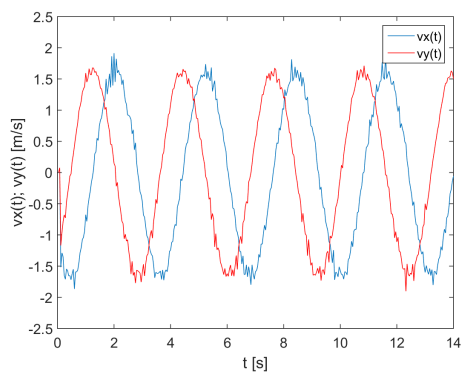
(b) Referencial global

Figura 6.42: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=0.9\text{rad/s}$, $d=2\text{m}$)

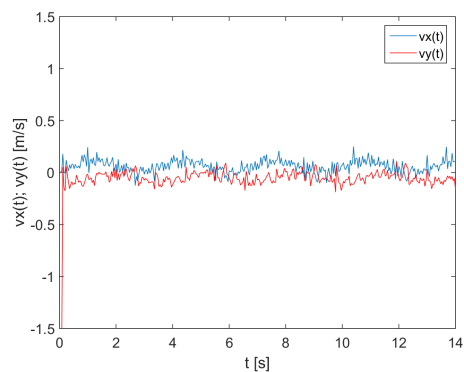
(a) Plano xOy 

(b) Em função do tempo

Figura 6.43: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=1.8\text{rad/s}$, $d=1\text{m}$)

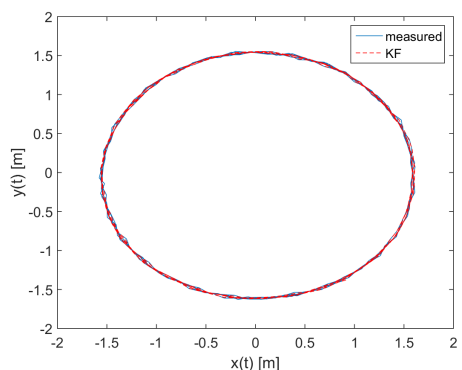
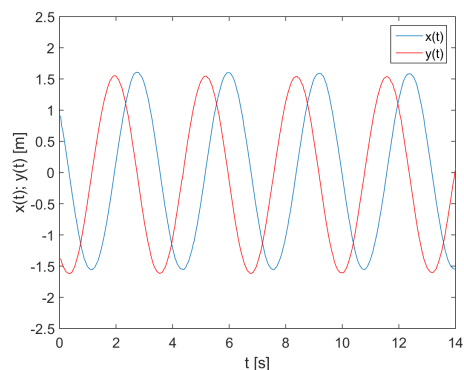


(a) Referencial local



(b) Referencial global

Figura 6.44: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=1.8\text{rad/s}$, $d=1\text{m}$)

(a) Plano xOy 

(b) Em função do tempo

Figura 6.45: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local ($\omega=1.8\text{rad/s}$, $d=2\text{m}$)

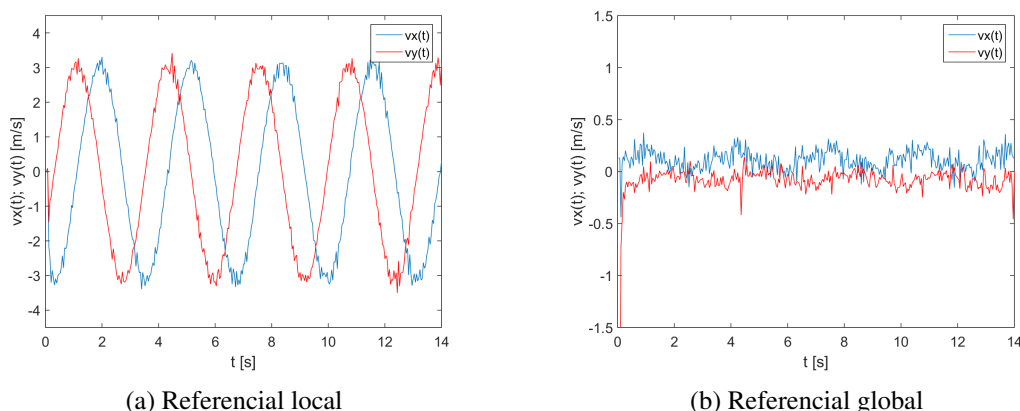


Figura 6.46: Ensaio 2 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola ($\omega=1.8\text{rad/s}$, $d=2\text{m}$)

As figuras 6.40a e 6.40b contêm as componentes x e y da velocidade estimada da bola em função do tempo, nos referenciais local e global, respetivamente. Pode-se verificar que a velocidade da bola no referencial local é representada por uma senoide cuja amplitude corresponde aproximadamente ao produto da velocidade de rotação do robô e da distância deste à bola. Convertendo esta velocidade para o referencial global pode-se verificar que esta de facto tende para zero.

No instante inicial, é visível um pico de velocidade devido ao facto de se ter inicializado o vetor de estado com velocidade 0m/s , no referencial local. Como esta se encontrava parada em relação ao referencial global e o robô encontrava-se a rodar a uma velocidade angular de 0.9rad/s , à distância de 1m , a velocidade real da bola é de 0.9m/s . Contudo, decidiu-se ter uma inicialização errada do filtro para testar e comprovar a rapidez de convergência do filtro.

Para avaliar a influência da distância de observação e da velocidade de rotação do robô na estimação de posição e velocidade da bola, realizaram-se três ensaios adicionais, de modo a que o robô estivesse a rodar a uma velocidade angular de 0.9rad/s a uma distância de 2m da bola (figuras 6.41 e 6.42), a rodar a uma velocidade angular de 1.8rad/s a uma distância de 1m da bola (figuras 6.43 e 6.44) e, finalmente, a rodar a uma velocidade angular de 1.8rad/s a uma distância de 2m da bola (figuras 6.45 e 6.46). A estimação foi realizada com a matriz de covariância do modelo de transição de estado Q2. A partir da análise destas figuras é visível que quanto maior é a velocidade de rotação do robô, e quando maior a distância deste à bola, maior vai ser o erro na estimativa de velocidade. Note-se que quanto maior é a distância do robô à bola, maior vai ser o erro entre a posição medida e a posição real da mesma, o que influencia negativamente a estimativa de velocidade.

Ensaio 3 - Robô Parado e Bola em Movimento

Nesta secção serão analisados os resultados do ensaio em que um robô estático observa uma bola em movimento maioritariamente segundo a direção x do referencial global, após um remate

rasteiro. Comece por se considerar que o filtro de Kalman utiliza a matriz de covariância do modelo de transição de estado Q1.

O gráfico da figura 6.47a contém as posições estimadas e medidas da bola no plano xOy , no referencial local. A figura 6.47b apresenta as componentes x e y da posição estimada da bola em função do tempo, no referencial local. A pequena componente de posição em y resulta de pequenos desalinhamentos do *kicker* com o centro da bola. Desde já pode-se observar que a posição estimada pelo filtro não descreveu completamente a trajetória da mesma.

As figuras 6.48a e 6.48b contém as componentes x e y da velocidade estimada da bola em função do tempo, nos referenciais local e global, respetivamente. À semelhança do ensaio presente na secção 6.4.5.2, o modelo utilizado para a predição de estado não descreve corretamente a dinâmica da bola, pelo que, é necessário dar uma maior importância às medidas para que o filtro consiga convergir para o valor correto.

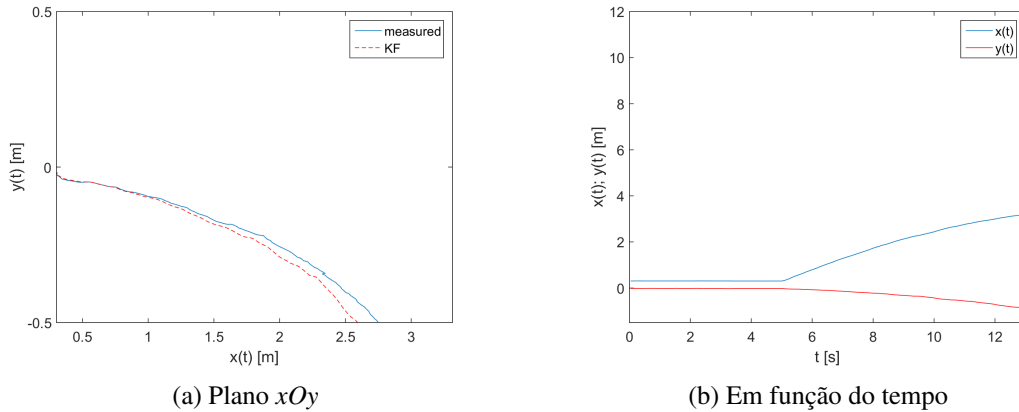


Figura 6.47: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q1) - posição da bola no referencial local

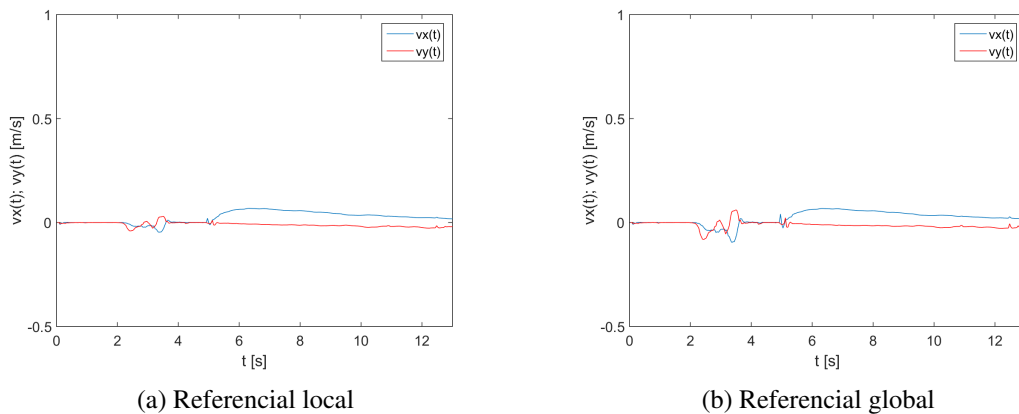


Figura 6.48: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q1) - velocidade da bola

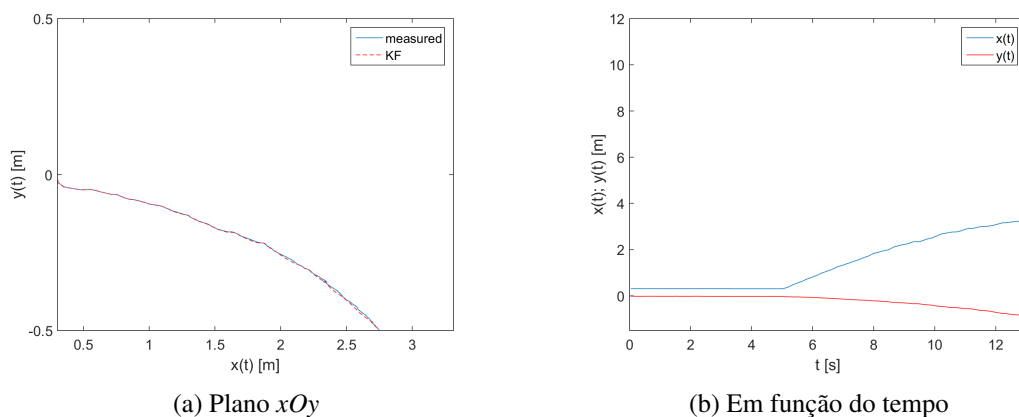


Figura 6.49: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local

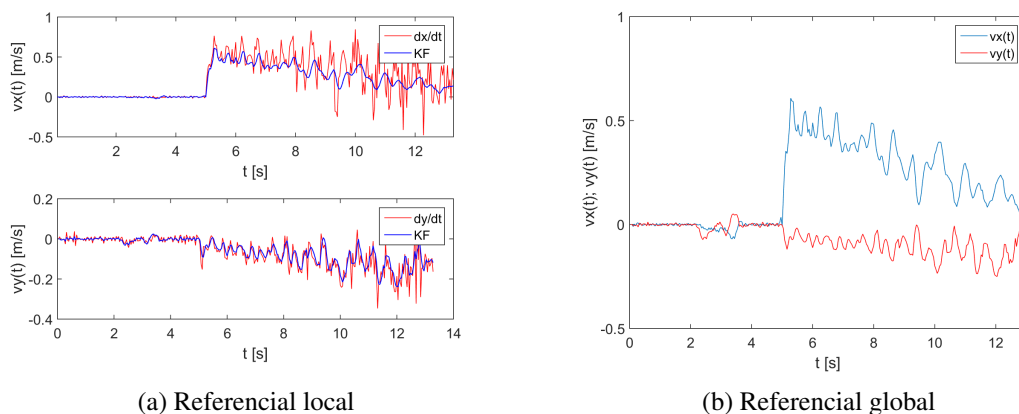


Figura 6.50: Ensaio 3 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola

Desta forma, realizou-se novamente a estimação de posição e velocidade com os mesmos dados de entrada, mas desta vez com uma matriz de covariância do modelo de transição de estado com variâncias $\sigma_{v_x}^2$ e $\sigma_{v_y}^2$ maiores, de modo a dar mais importância às medidas da bola (matriz Q2, equação (6.52)).

As figuras 6.49 e 6.50 contêm o resultado dos ensaios para a nova matriz de covariância. Analisando a figura 6.50a, pode-se verificar que a velocidade estimada já foi de encontro ao valor esperado, porém, dando uma maior importância às medidas resulta num maior ruído na estimativa. Mesmo assim, a velocidade estimada pelo filtro de Kalman contém menos ruído do que a estimativa calculada a partir da derivada da posição em função do tempo. O traçado da velocidade estimada no referencial global é idêntico à velocidade no referencial local devido ao facto de o robô se encontrar parado e com uma orientação de 0° em relação ao referencial global. Pode-se também observar que, embora não se tenha dado maior importância às observações para a estimação da posição da bola, esta já foi de encontro à trajetória da mesma, devido à correta estimativa de velocidade.

Ensaio 4 - Bola Contra Robô

Nesta secção será apresentado um ensaio adicional, em que uma bola foi lançada em direção a um robô estacionário a observar a mesma, de modo a que a colisão originasse uma mudança brusca de direção da trajetória da bola. Com este ensaio pretende-se testar a capacidade do filtro conseguir detetar uma variação brusca na velocidade do alvo que está a seguir. Para obter uma boa estimativa da posição e velocidade da bola e, tendo em conta os resultados obtidos na secção 6.4.5.3, decidiu-se utilizar a matriz de covariância do modelo de transição de estado Q2 (equação (6.52)).

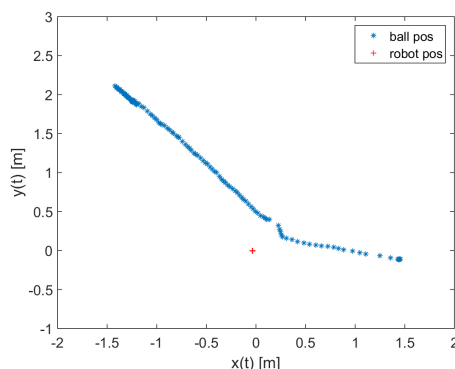
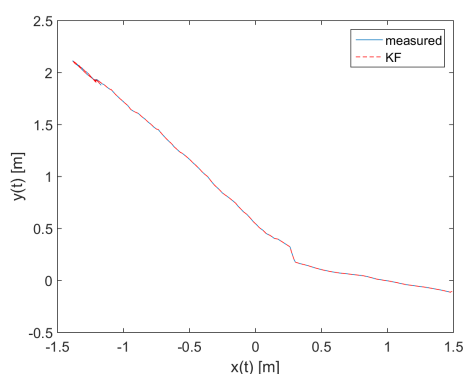
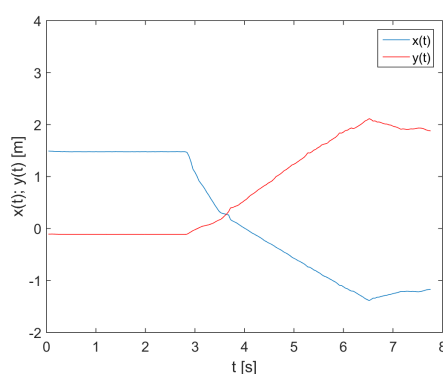


Figura 6.51: Ensaio 4 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - Posições da bola e do robô, no referencial global



(a) Plano xOy



(b) Em função do tempo

Figura 6.52: Ensaio 4 do filtro de Kalman com vetor de estado no referencial local com dados obtidos num robô real (Q2) - posição da bola no referencial local

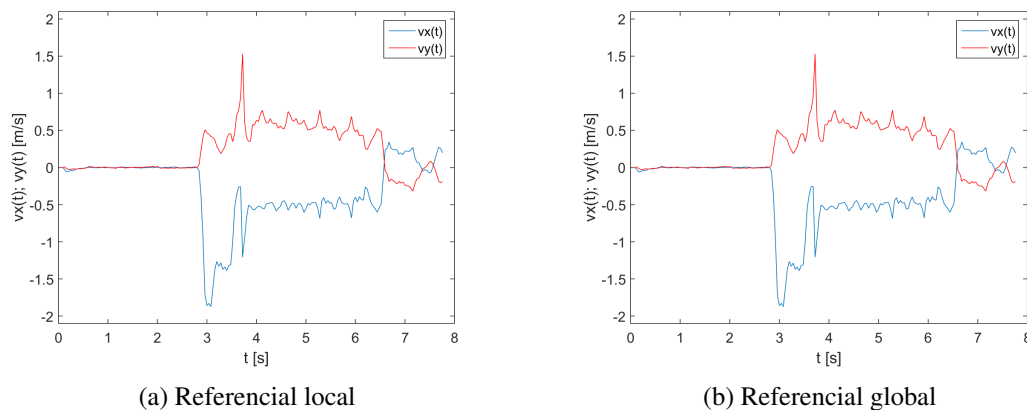


Figura 6.53: Ensaio 4 do filtro de Kalman com vetor de estado no referencial local, com dados obtidos num robô real (Q2) - velocidade da bola

A figura 6.51 apresenta uma representação do cenário do ensaio de vista de topo, contendo a posição do robô que realiza a observação e as posições estimadas da bola, no referencial global. A figura 6.52a contém a posição estimada da bola, no plano xOy do referencial local. Na figura 6.52b são visíveis as componentes x e y da posição estimada da bola em função do tempo, no referencial local. Como se pode verificar, este estimador conseguiu estimar com precisão a posição da bola. Note-se que os traçados das figuras 6.51 e 6.52a são semelhantes porque o robô se encontrava parado e com uma orientação de 0° em relação ao referencial global.

As figuras 6.53a e 6.53b contêm as componentes x e y da velocidade da bola em função do tempo, nos referenciais local e global, respetivamente. Até ao instante de tempo $t=3s$, a bola encontrava-se parada. De seguida, a bola é lançada em direção ao robô, sendo visível as componentes de velocidade em x e y descritas pela bola. Aproximadamente em $t=6.5s$, a bola embate contra o robô, mudando de direção e perdendo velocidade durante a colisão.

6.5 Filtro de Kalman - Referencial Global

Na secção 6.4 foi apresentada uma abordagem de estimação de posição e velocidade de uma bola num referencial móvel com recurso a um filtro de Kalman, cujo vetor de estado continha a posição e velocidade da bola no referencial local de cada robô. Para efeitos comparativos e de validação de resultados, desenvolveu-se um segundo estimador que também consiste num filtro de Kalman, todavia, a matriz de transição de estado será composta pela posição e velocidade da bola no referencial global (do mundo).

Desta forma, começará por ser apresentado o dimensionamento do filtro de Kalman e, de seguida, será apresentado um conjunto de ensaios ao filtro com dados gerados num *script* Matlab, obtidos em ambiente de simulação e através de um robô real. Para além disso, serão também apresentados resultados aos ensaios realizados a uma implementação deste filtro, realizada no *software* de decisão dos agentes da equipa 5DPO.

6.5.1 Dimensionamento do Filtro

Comece por se definir a matriz de estado X como:

$$X(k) = \begin{bmatrix} x_b^W(k) \\ y_b^W(k) \\ v_{x_b}^W(k) \\ v_{y_b}^W(k) \end{bmatrix} \quad (6.53)$$

Numa primeira fase, o filtro de Kalman realiza a previsão do estado da bola (6.54), atualizando a matriz de covariância do erro associado a esta estimativa (6.32).

$$\hat{X}^-(k) = A(k)\hat{X}(k-1) \quad (6.54)$$

Tendo por base o modelo de transição de estado apresentado em 6.15, a equação de transição de estado do filtro de Kalman fica dada por 6.55.

$$\begin{bmatrix} x_b^W(k) \\ y_b^W(k) \\ v_{x_b}^W(k) \\ v_{y_b}^W(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{1-e^{-C_1T}}{C_1} & 0 \\ 0 & 1 & 0 & \frac{1-e^{-C_1T}}{C_1} \\ 0 & 0 & e^{-C_1T} & 0 \\ 0 & 0 & 0 & e^{-C_1T} \end{bmatrix} \begin{bmatrix} x_b^W(k-1) \\ y_b^W(k-1) \\ v_{x_b}^W(k-1) \\ v_{y_b}^W(k-1) \end{bmatrix} \quad (6.55)$$

Consequentemente, a matriz de transição de estado fica:

$$A = \begin{bmatrix} 1 & 0 & \frac{1-e^{-C_1T}}{C_1} & 0 \\ 0 & 1 & 0 & \frac{1-e^{-C_1T}}{C_1} \\ 0 & 0 & e^{-C_1T} & 0 \\ 0 & 0 & 0 & e^{-C_1T} \end{bmatrix} \quad (6.56)$$

Analisando a equação de transição de estado (6.55), é possível verificar que o sistema é linear, comprovando que se pode utilizar um filtro de Kalman convencional. Desta forma, as equações de covariância do erro de estimação de estado, atualização do ganho do filtro, correção de estado e de covariância do erro são dadas, respetivamente, por (6.32), (6.38), (6.39) e (6.40).

Na fase de correção, o filtro de Kalman utiliza as medidas da bola realizadas pelo robô que está a realizar a observação. A matriz de estado está referenciada num referencial global (do mundo), porém, a observação é realizada no referencial local do robô (x_{bm}^R, y_{bm}^R). Desta forma, é necessário aplicar uma rotação seguida duma translação para converter as coordenadas da posição da bola do referencial local para o referencial global. Definindo x_r e y_r como as coordenadas do robô no referencial global, $Z \in \mathbb{R}^2$ é dado pela seguinte equação:

$$Z(k) = \begin{bmatrix} x_{bm}^W(k) \\ y_{bm}^W(k) \end{bmatrix} = R_{z,\theta} \begin{bmatrix} x_{bm}^R(k) \\ y_{bm}^R(k) \end{bmatrix} + \begin{bmatrix} x_r(k) \\ y_r(k) \end{bmatrix} \quad (6.57)$$

onde

$$R_{z,\theta(k)} = \begin{bmatrix} \cos(\theta(k)) & -\sin(\theta(k)) \\ \sin(\theta(k)) & \cos(\theta(k)) \end{bmatrix} \quad (6.58)$$

Consequentemente, a matriz $H \in \mathbb{R}^{2 \times 4}$ é dada por (6.42) e, a matriz do ganho do filtro de Kalman K fica definida em $\mathbb{R}^{4 \times 2}$. Por sua vez, a matriz da covariância do erro $P \in \mathbb{R}^{4 \times 4}$.

À semelhança do que foi apresentado na secção 6.4.2, o ruído de medição foi obtido através de medições da bola realizadas a diferentes distâncias de observação, e com robôs reais. Devido à influência da distância do robô à bola com a incerteza da observação, adotou-se o modelo de ruído variável semelhante ao apresentado na secção 6.4.3, também utilizado na secção anterior.

Contudo, como se pode verificar na equação 6.57, a posição da bola no referencial global é uma função de cinco variáveis $h(\cdot) = f(x_b^R, y_b^R, x_r, y_r, \theta_r)$, mais concretamente, é uma função que depende da observação da bola no referencial local e da pose do mesmo no referencial global. Como estas variáveis podem ter incertezas diferentes, é necessário ter esse aspeto em consideração no cálculo da covariância da localização da bola no referencial global.

Considere-se que as observações da bola no referencial local e a posição do mesmo em cada instante de tempo são dadas por distribuições normais, com matrizes de covariância Σ_b^R e Σ_r , respetivamente, tal que:

$$\Sigma_b^R = \begin{bmatrix} \sigma_{x_b^R x_b^R}^2 & \sigma_{x_b^R y_b^R}^2 \\ \sigma_{y_b^R x_b^R}^2 & \sigma_{y_b^R y_b^R}^2 \end{bmatrix} \quad (6.59)$$

$$\Sigma_r = \begin{bmatrix} \sigma_{x_r x_r}^2 & \sigma_{x_r y_r}^2 & \sigma_{x_r \theta_r}^2 \\ \sigma_{y_r x_r}^2 & \sigma_{y_r y_r}^2 & \sigma_{y_r \theta_r}^2 \\ \sigma_{\theta_r x_r}^2 & \sigma_{\theta_r y_r}^2 & \sigma_{\theta_r \theta_r}^2 \end{bmatrix} \quad (6.60)$$

Defina-se ainda Σ e Σ_b^W tal que:

$$\Sigma = \begin{bmatrix} \Sigma_b^R & 0 \\ 0 & \Sigma_r \end{bmatrix} = \begin{bmatrix} \sigma_{x_b^R x_b^R}^2 & \sigma_{x_b^W y_b^W}^2 & 0 & 0 & 0 \\ \sigma_{y_b^R x_b^R}^2 & \sigma_{y_b^W y_b^W}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{x_r x_r}^2 & \sigma_{x_r y_r}^2 & \sigma_{x_r \theta_r}^2 \\ 0 & 0 & \sigma_{y_r x_r}^2 & \sigma_{y_r y_r}^2 & \sigma_{y_r \theta_r}^2 \\ 0 & 0 & \sigma_{\theta_r x_r}^2 & \sigma_{\theta_r y_r}^2 & \sigma_{\theta_r \theta_r}^2 \end{bmatrix} \quad (6.61)$$

Desta forma, o jacobiano fica definido por:

$$\nabla h = \begin{bmatrix} x_b^W(k) \\ y_b^W(k) \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x_b^R} & \frac{\partial}{\partial y_b^R} & \frac{\partial}{\partial x_r} & \frac{\partial}{\partial y_r} & \frac{\partial}{\partial \theta_r} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_b^W}{\partial x_b^R} & \frac{\partial x_b^W}{\partial y_b^R} & \frac{\partial x_b^W}{\partial x_r} & \frac{\partial x_b^W}{\partial y_r} & \frac{\partial x_b^W}{\partial \theta_r} \\ \frac{\partial y_b^W}{\partial x_b^R} & \frac{\partial y_b^W}{\partial y_b^R} & \frac{\partial y_b^W}{\partial x_r} & \frac{\partial y_b^W}{\partial y_r} & \frac{\partial y_b^W}{\partial \theta_r} \end{bmatrix} \quad (6.62)$$

Calculando as derivadas parciais:

$$\frac{\partial x_b^W}{\partial x_b^R} = \cos(\theta_r) \quad (6.63)$$

$$\frac{\partial y_b^W}{\partial x_b^R} = \sin(\theta_r) \quad (6.64)$$

$$\frac{\partial x_b^W}{\partial y_b^R} = -\sin(\theta_r) \quad (6.65)$$

$$\frac{\partial y_b^W}{\partial y_b^R} = \cos(\theta_r) \quad (6.66)$$

$$\frac{\partial x_b^W}{\partial x_r} = 1 \quad (6.67)$$

$$\frac{\partial y_b^W}{\partial x_r} = 0 \quad (6.68)$$

$$\frac{\partial x_b^W}{\partial y_r} = 0 \quad (6.69)$$

$$\frac{\partial y_b^W}{\partial y_r} = 1 \quad (6.70)$$

$$\frac{\partial x_b^W}{\partial \theta_r} = -x_b^R \sin(\theta_r) - y_b^R \cos(\theta_r) \quad (6.71)$$

$$\frac{\partial y_b^W}{\partial \theta_r} = x_b^R \cos(\theta_r) - y_b^R \sin(\theta_r) \quad (6.72)$$

Segundo a lei de propagação do erro [67]:

$$R = \nabla h \sum \nabla h^T = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) & 1 & 0 & -x_b^R \sin(\theta_r) - y_b^R \cos(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) & 0 & 1 & x_b^R \cos(\theta_r) - y_b^R \sin(\theta_r) \end{bmatrix} \cdot \begin{bmatrix} \sigma_{x_b^R x_b^R}^2 & \sigma_{x_b^R y_b^R}^2 & 0 & 0 & 0 \\ \sigma_{y_b^R x_b^R}^2 & \sigma_{y_b^R y_b^R}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{x_r x_r}^2 & \sigma_{x_r y_r}^2 & \sigma_{x_r \theta_r}^2 \\ 0 & 0 & \sigma_{y_r x_r}^2 & \sigma_{y_r y_r}^2 & \sigma_{y_r \theta_r}^2 \\ 0 & 0 & \sigma_{\theta_r x_r}^2 & \sigma_{\theta_r y_r}^2 & \sigma_{\theta_r \theta_r}^2 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_r) & \sin(\theta_r) \\ -\sin(\theta_r) & \cos(\theta_r) \\ 1 & 0 \\ 0 & 1 \\ -x_b^R \sin(\theta_r) - y_b^R \cos(\theta_r) & x_b^R \cos(\theta_r) - y_b^R \sin(\theta_r) \end{bmatrix} \quad (6.73)$$

A obtenção do ruído do modelo de transição de estado não é tão trivial, devido ao facto de a bola estar num ambiente bastante dinâmico, onde o movimento livre da bola pode ser interrompido a qualquer momento por uma colisão com um ou mais robôs, cuja intensidade do choque é também desconhecida. Consequentemente, a dinâmica da bola fica imprevisível, sendo impossível analisar com rigor as forças que atuam na mesma. Como tal, optou-se por utilizar esta matriz como parâmetro de afinação do filtro, realizando-se ensaios com diferentes valores dentro de ordens de grandeza próximas à do ruído de medição. Considerando $\sigma_{q_x}^2$, $\sigma_{q_y}^2$, $\sigma_{q_{v_x}}^2$ e $\sigma_{q_{v_y}}^2$ como as variâncias do ruído do processo de processo em x , y , v_x e v_y , a matriz $Q \in \mathbb{R}^{4 \times 4}$ fica definida por:

$$Q = \begin{bmatrix} \sigma_{q_x}^2 & 0 & 0 & 0 \\ 0 & \sigma_{q_y}^2 & 0 & 0 \\ 0 & 0 & \sigma_{q_{v_x}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{q_{v_y}}^2 \end{bmatrix} \quad (6.74)$$

6.5.2 Ensaios Realizados

Para validar e avaliar o desempenho deste filtro de Kalman, realizou-se o mesmo conjunto de ensaios referidos na secção 6.4.5 (figura 6.8), testando, assim, várias combinações de estados do robô e da bola.

De seguida, serão apresentados os ensaios realizados, tendo como base dados originados de três fontes distintas, concretamente, pose do robô e observação da bola gerados em Matlab, obtidos em ambiente de simulação e, finalmente, obtidos a partir de *logs* tirados em robôs reais. Para se obter um meio de comparação, os dados utilizados nestes ensaios são os mesmos que os utilizados nos ensaios do filtro de Kalman com o vetor de estado referenciado no referencial local do robô. Para não tornar esta dissertação muito extensa, serão apenas apresentados os resultados que demonstrem diferenças evidentes, ou que mereçam algum realce em relação aos resultados obtidos através do filtro de Kalman no referencial local.

Embora o filtro de Kalman estime a posição e velocidade da bola no referencial global, por vezes serão apresentadas as posições e velocidades no referencial local do robô através das equações (6.75) e (6.76).

$$\begin{bmatrix} x_b^R(k) \\ y_b^R(k) \end{bmatrix} = R_{z, -\theta(k)} \left(\begin{bmatrix} x_b^W(k) \\ y_b^W(k) \end{bmatrix} - \begin{bmatrix} x_r(k) \\ y_r(k) \end{bmatrix} \right) \quad (6.75)$$

$$\begin{bmatrix} v_{x_b}^R(k) \\ v_{y_b}^R(k) \end{bmatrix} = R_{z, -\theta(k)} \begin{bmatrix} v_{x_b}^W(k) \\ v_{y_b}^W(k) \end{bmatrix} - \begin{bmatrix} v(k) \\ v_n(k) \end{bmatrix} - d(k)\omega(k) \begin{bmatrix} -\sin(\alpha(k)) \\ \cos(\alpha(k)) \end{bmatrix} \quad (6.76)$$

onde

$$R_{z, -\theta(k)} = \begin{bmatrix} \cos(\theta(k)) & \sin(\theta(k)) \\ -\sin(\theta(k)) & \cos(\theta(k)) \end{bmatrix} \quad (6.77)$$

6.5.2.1 Ensaios com Dados Gerados no Matlab

Nos ensaios apresentados nas próximas secções, as medidas da bola e as posições e velocidades do robô que realiza a observação foram geradas a partir de um *script* em Matlab. Numa primeira fase, assumir-se-ão cenários em que o ruído é inexistente e, de seguida serão analisadas observações com ruído de medição e de localização do robô na mesma ordem de grandeza do presente nos robôs da equipa 5DPO.

Para o cálculo da matriz de covariância do erro do modelo de transição de estado Q , utilizou-se um método semelhante ao utilizado na implementação anterior. Uma primeira matriz utilizada nos ensaios é a matriz $Q1$, dada pela equação (6.78). Assim que for relevante, serão apresentadas e devidamente identificadas as alterações a esta matriz.

$$Q1 = \begin{bmatrix} 3.7 \times 10^{-6} & 0 & 0 & 0 \\ 0 & 3.7 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 3.7 \times 10^{-6} & 0 \\ 0 & 0 & 0 & 3.7 \times 10^{-6} \end{bmatrix} \quad (6.78)$$

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

Esta secção corresponde ao ensaio 1 apresentado na secção 6.4.5.1. O filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção, apresentando resultados bastante similares. Como tal, serão apenas apresentadas as figuras que se apresentaram ligeiras diferenças que se devem realçar.

A figura 6.54 pretende ilustrar o cenário de ensaio, contendo a trajetória descrita pelo robô e as posições estimadas da bola no plano xOy do referencial global. A figura 6.55a contém as posições medidas e a conversão das estimativas da bola, no plano xOy do referencial local do robô. A figura 6.55b apresenta as componentes x e y da conversão da posição estimada da bola em função do tempo, do referencial global para referencial local. Inicializou-se propositadamente o vetor de estado com uma posição com o desvio de 1m em relação à primeira observação do robô, em ambas as componentes x e y . Com isto, pretende-se testar e demonstrar a rapidez de convergência deste estimador.

As figuras 6.56a e 6.56b contêm as componentes x e y da velocidade estimada da bola em função do tempo, nos referenciais local e global, respetivamente. A velocidade estimada no referencial global tende para 0m/s, como seria de esperar. O pico visível na figura 6.56b deve-se ao facto de o vetor de estado ter sido inicializado com velocidades v_x e v_y iguais a 1m/s. Como este vetor está referenciado no referencial global e a bola está parada em relação a este referencial, o ideal seria inicializar o vetor de estado com velocidades iguais a 0m/s. No entanto, decidiu-se inicializar este vetor com uma velocidade errada de modo a avaliar a rapidez de convergência do

estimador. Embora a velocidade de convergência seja rápida, pode-se constatar que esta é ligeiramente mais lenta que a do filtro com o vetor de estado referenciado no referencial local (figura 6.11b). Convertendo a velocidade estimada para o referencial local, verifica-se que esta é representada por uma onda quadrada de amplitude igual a 1m/s, coincidindo a velocidade linear com que se deslocava o robô que estava a efetuar a observação. Ainda em relação ao ensaio 1 da secção 6.4.5.1, é visível que este estimador já não origina as pequenas oscilações verificadas na figura 6.11b. Isto deve-se ao facto de a bola estar referenciada no referencial global, fazendo com que ela apresente uma dinâmica muito mais suave (porque está parada neste referencial). Desta forma, a estimativa de velocidade deixa de ser afetada pelos erros de discretização.

A figura 6.57 contém os traçados da velocidade estimada da bola através da aproximação numérica (diferença entre duas posições sucessivas) da derivada da posição em função do tempo, utilizando as posições medidas e estimadas pelo filtro.

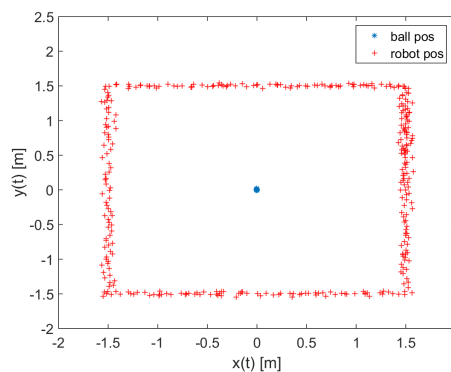
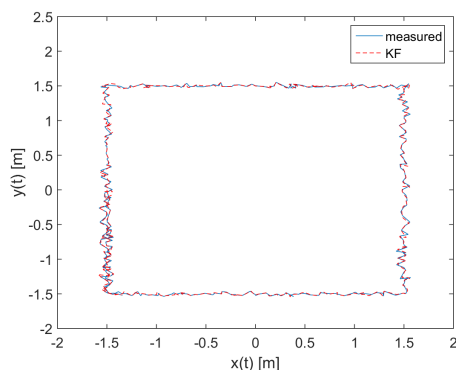
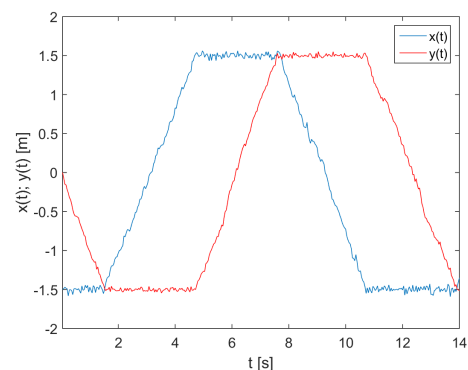


Figura 6.54: Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados gerados em Matlab, com ruído (Q1) - Posições da bola e do robô, no referencial global



(a) Plano xOy



(b) Em função do tempo

Figura 6.55: Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados gerados em Matlab, com ruído (Q1) - Posição da bola no referencial local

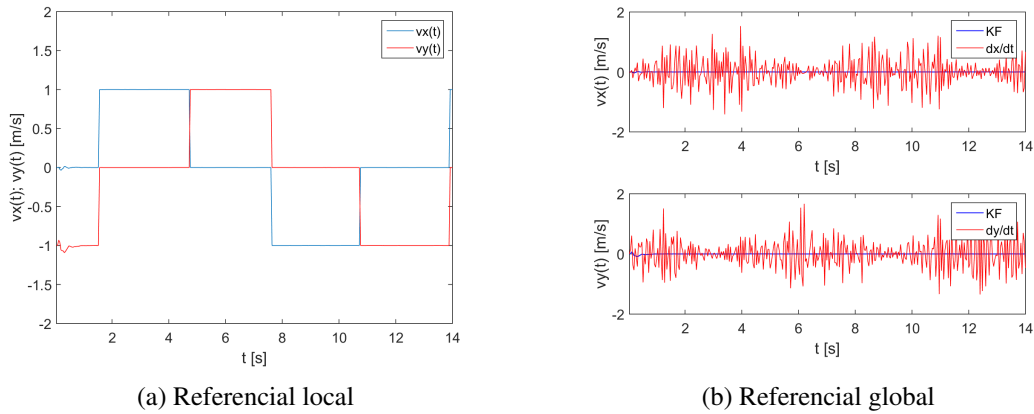


Figura 6.56: Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados gerados em Matlab, com ruído (Q1) - velocidade da bola

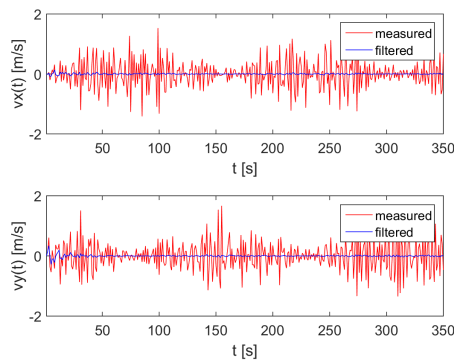


Figura 6.57: Comparação da velocidade estimada através da aproximação numérica da derivada da posição em função do tempo, utilizando as posições medidas ou utilizando as posições estimadas pelo filtro de Kalman

À semelhança do que se verificou para as posições estimadas pelo filtro de Kalman no referencial local, pode-se verificar que através da utilização das posições estimadas, consegue-se obter traçados com menor ruído do que através da utilização direta das medidas. No entanto, esta forma de calcular a velocidade tem mais ruído do que a estimativa da velocidade feita pelo filtro, como se pode verificar comparando a figura 6.57 com 6.56.

Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

Esta secção corresponde ao ensaio 2 apresentado na secção 6.4.5.1. O filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção, não se tendo verificado nenhuma diferença evidente a destacar.

Ensaio 3 - Robô Parado e Bola em Movimento

Esta secção corresponde ao ensaio 3 apresentado na secção 6.4.5.1. O filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção, não se tendo verificado nenhuma diferença evidente a destacar.

6.5.2.2 Ensaios com Dados Obtidos no Simulador

Nas próximas secções serão apresentados os ensaios descritos em 6.4.5, em que os dados de entrada do filtro de Kalman foram obtidos através de *logs* retirados no *software* de decisão da equipa 5DPO, recorrendo a cenários simulados em SimTwo. Desta forma, as velocidades e ações do robô são definidas a partir do *Decision*, enquanto que as posições do robô e da bola são obtidas no simulador, possuindo o simulador um motor de física que simula a dinâmica do robô e da bola.

Para tornar a simulação mais realista, foi adicionado, no simulador, ruído à observação da bola por parte dos agentes na mesma ordem de grandeza do ruído existente nos robôs reais. Este ruído tinha em conta a distância a que o robô se encontrava da bola, de acordo com as expressões (6.45), (6.46) e (6.47).

Durante os ensaios foi utilizada a mesma matriz de covariância do modelo de transição de estados utilizada na secção 6.5.2.1, nomeadamente, dada pela matriz Q1 (equação (6.78)).

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

Esta secção corresponde ao ensaio 1 apresentado na secção 6.4.5.2. O filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção, não se tendo verificado nenhuma diferença evidente a destacar.

Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

Esta secção corresponde ao ensaio 2 apresentado na secção 6.4.5.2. O filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção, não se tendo verificado nenhuma diferença evidente a destacar.

Ensaio 3 - Robô Parado e Bola em Movimento

Esta secção corresponde ao ensaio 3 apresentado na secção 6.4.5.2, sendo que o filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção. Como os resultados deste ensaio evidenciaram algumas diferenças em relação aos presentes na referida secção, estes serão apresentados de seguida.

O gráfico da figura 6.58a apresenta as posições medidas e a conversão das posições estimadas da bola no plano xOy , do referencial local. A figura 6.58b contém as componentes x e y da posição estimada da bola em função do tempo, no referencial local.

As figuras 6.59a e 6.59b contêm as componentes x e y da velocidade estimada da bola em função do tempo, nos referenciais local e global, respetivamente. Para além disso, na figura 6.59a adicionou-se uma estimativa de velocidade através da derivada das posições estimadas pelo filtro em função do tempo.

O traçado da velocidade no referencial local assemelha-se ao do referencial global devido ao facto de o robô se encontrar parado, com uma orientação de 0° em relação ao eixo do x do referencial global.

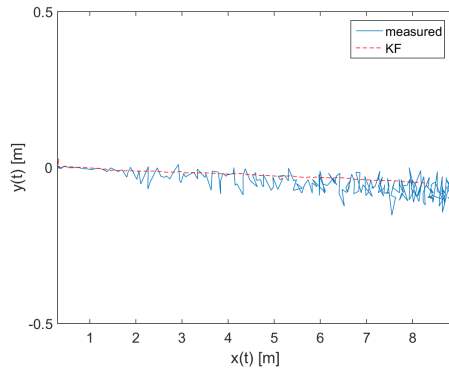
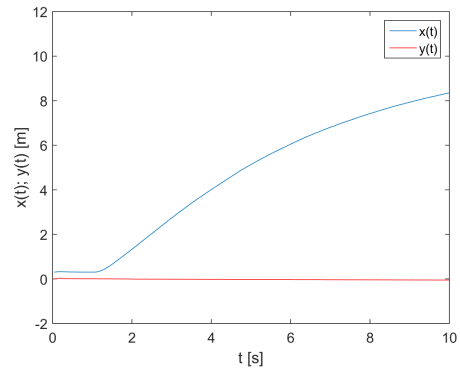
Observando a figura 6.59b, verifica-se que o valor estimado pelo filtro se encontrou longe do valor real. Isto deve-se ao facto de o modelo a dinâmica da bola utilizado pelo filtro ser uma aproximação do modelo real, pelo facto de o modelo simulado pelo SimTwo ser bastante mais realista, assim como pelo facto de a bola ter uma dinâmica muito mais rápida e difícil de acompanhar com os valores utilizados para a matriz Q . Tal efeito explica também o pequeno desvio da conversão das posições estimadas pelo filtro pelo valor médio das medidas pelo robô (figura 6.58a). Desta forma, tem de se dar maior importância às medidas, de modo a que estas consigam "corrigir" a diferença entre os modelos. Assim, aumentaram-se as variâncias em v_x e v_y , obtendo-se a matriz $Q2$ presente na equação (6.79):

$$Q2 = \begin{bmatrix} 3.7 \times 10^{-6} & 0 & 0 & 0 \\ 0 & 3.7 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 3.7 \times 10^{-4} & 0 \\ 0 & 0 & 0 & 3.7 \times 10^{-4} \end{bmatrix} \quad (6.79)$$

As figuras 6.60 e 6.61 apresentam os resultados do ensaio para a nova matriz de covariância $Q2$. Comparando com a figura 6.59b, pode-se verificar que, dando mais importância às medidas, já se consegue obter uma estimativa de velocidade mais perto do valor real, contudo, ainda com um erro elevado. Porém, verifica-se que com uma melhor estimativa de velocidade, a conversão da posição estimada pelo filtro do referencial global para o local já se aproxima mais da trajetória real da bola.

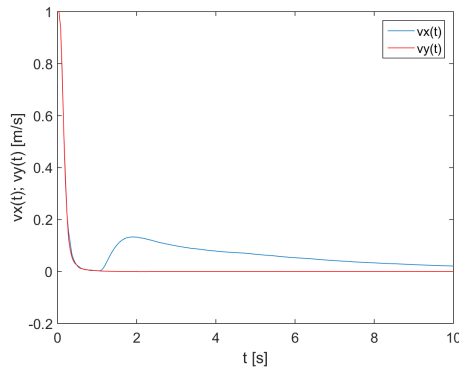
Para tentar melhorar a estimação de velocidade da bola aumentou-se novamente as variâncias σ_{v_x} e σ_{v_y} , dando ainda mais importância às medidas. Como tal, realizou-se novamente a estimação, para o mesmo conjunto de dados de entrada do filtro, mas com uma matriz de covariância do erro de transição de estado $Q3$, presente na equação (6.80).

$$Q3 = \begin{bmatrix} 3.7 \times 10^{-6} & 0 & 0 & 0 \\ 0 & 3.7 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 3.7 \times 10^{-2} & 0 \\ 0 & 0 & 0 & 3.7 \times 10^{-2} \end{bmatrix} \quad (6.80)$$

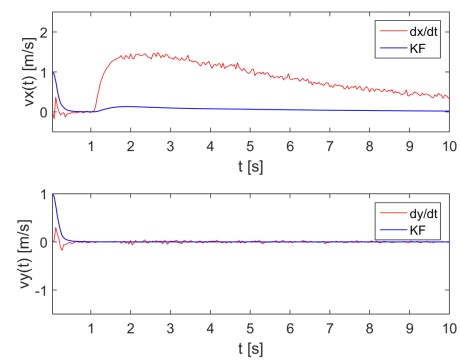
(a) Plano xOy 

(b) Em função do tempo

Figura 6.58: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q1) - Posição da bola no referencial local

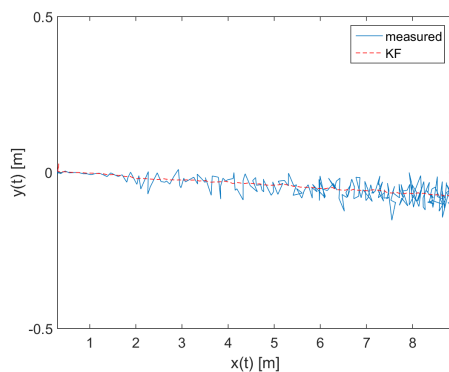
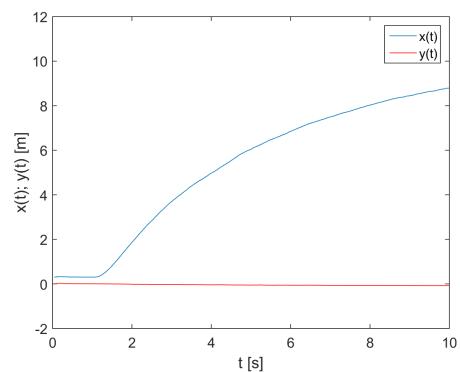


(a) Referencial local



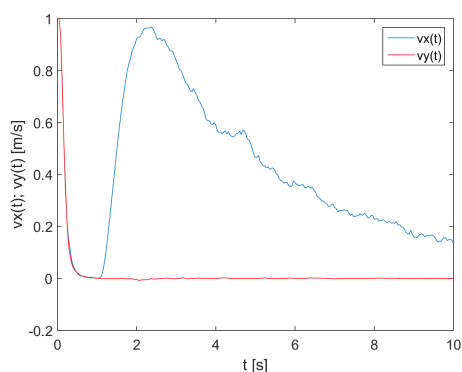
(b) Referencial global

Figura 6.59: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q1) - velocidade da bola

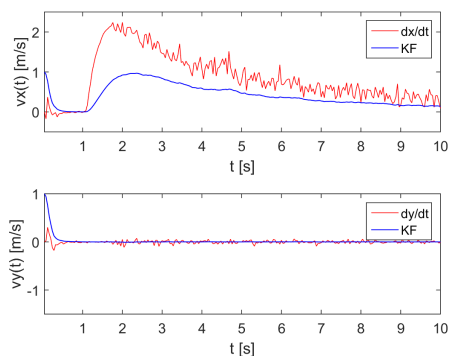
(a) Plano xOy 

(b) Em função do tempo

Figura 6.60: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q2) - Posição da bola no referencial local

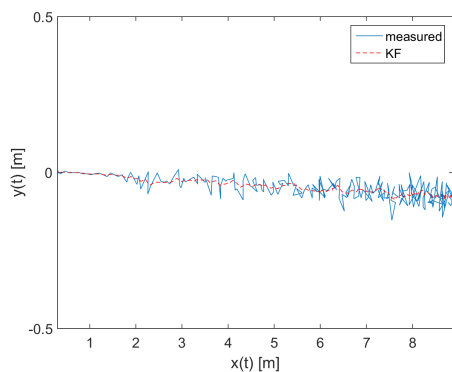
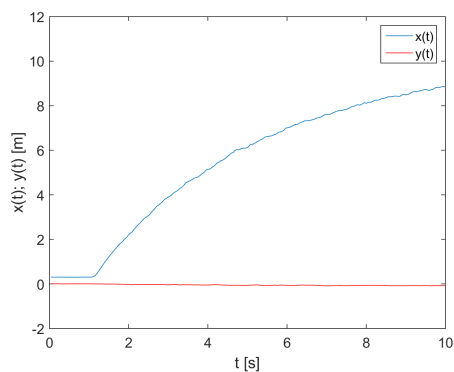


(a) Referencial local



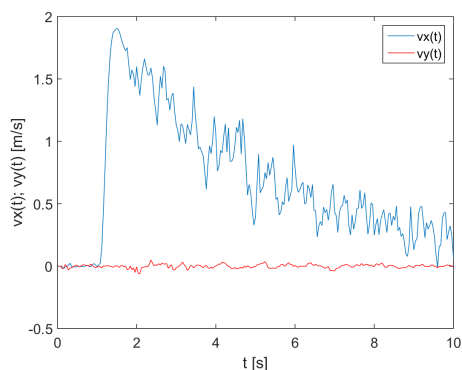
(b) Referencial global

Figura 6.61: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q2) - velocidade da bola

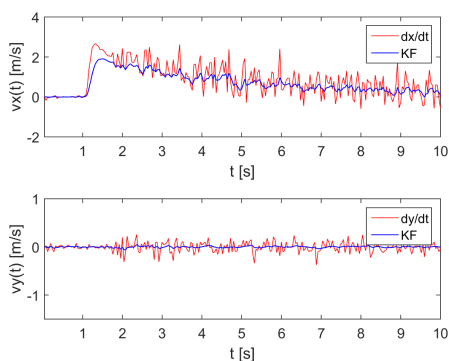
(a) Plano xOy 

(b) Em função do tempo

Figura 6.62: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q3) - Posição da bola no referencial local



(a) Referencial local



(b) Referencial global

Figura 6.63: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos em SimTwo (Q3) - velocidade da bola

As figuras 6.62 e 6.63 contêm os resultados do ensaio para a matriz de covariância Q3. Como se pode observar, dando mais importância às medidas já se consegue chegar a um valor absoluto de velocidade da bola muito próximo da velocidade real da mesma. Desta forma, é altamente recomendável que se utilize uma matriz de covariância Q de ordem de grandeza semelhantes à de Q3 nos instantes em que se sabe que a bola está em movimento, por exemplo, após um remate. Passado algum tempo, o ideal seria comutar o valor de Q, para que se voltasse novamente a filtrar mais as estimativas.

6.5.2.3 Ensaio com Dados Obtidos num Robô Real

Nas secções que se seguem serão apresentados os ensaios deste filtro de Kalman com os dados recolhidos em ambiente real, num robô da equipa 5PDO. Os dados foram armazenados num ficheiro de texto e, posteriormente, importados como entradas numa implementação deste filtro de Kalman em Matlab.

Visto que no ensaio 6.5.2.2 se verificou que, pelo menos no simulador, era necessário dar mais importância às observações da bola na estimativa da sua velocidade, decidiu-se começar por realizar os próximos ensaios utilizando a matriz de covariância Q2, dada por (6.79).

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

Esta secção corresponde ao ensaio 1 apresentado na secção 6.4.5.3, sendo que o filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção.

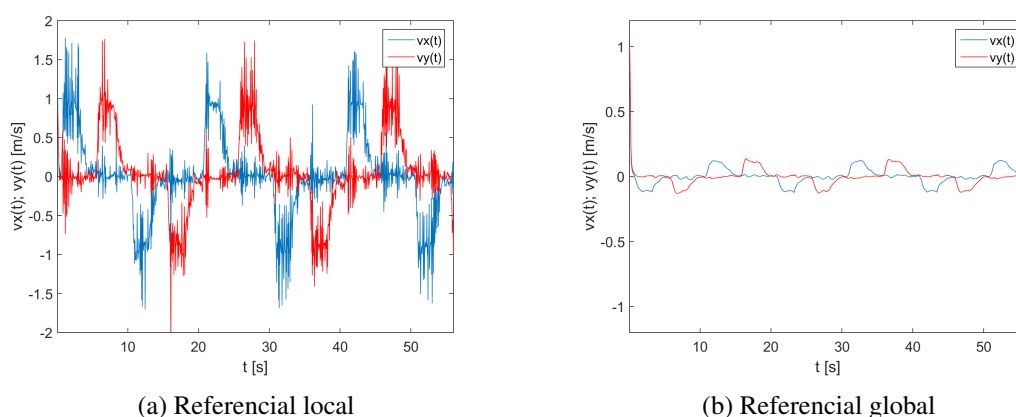


Figura 6.64: Ensaio 1 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos num robô real (Q2) - velocidade da bola

A estimativa de posição do filtro foi semelhante à obtida com o filtro de Kalman no referencial local. Porém, como nesta abordagem o vetor de estado está referenciada no referencial global, a

sua dinâmica é diferente, resultando numa estimativa de velocidade mais imune ao ruído, principalmente o ruído vindo da odometria, que prejudica a estimativa do filtro no referencial local. Por outro lado, a conversão da velocidade estimada da bola do referencial global para o local continua a ser afetada por este ruído. As figuras 6.64a e 6.64b demonstram os efeitos apresentados. Mais uma vez, pode-se verificar um elevado nível de ruído na figura 6.64a devido principalmente ao ruído anormalmente elevado nas medidas obtidas na odometria. Não é possível filtrar este ruído da mesma forma que se filtra o ruído das medidas, devido ao facto de este ser injetado diretamente aquando da conversão da velocidade do referencial global para o referencial local.

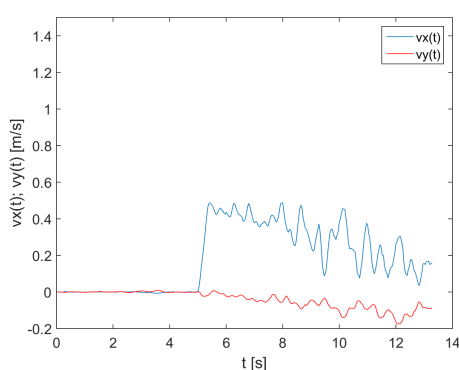
Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

Esta secção corresponde ao ensaio 2 apresentado na secção 6.4.5.3, sendo que o filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção.

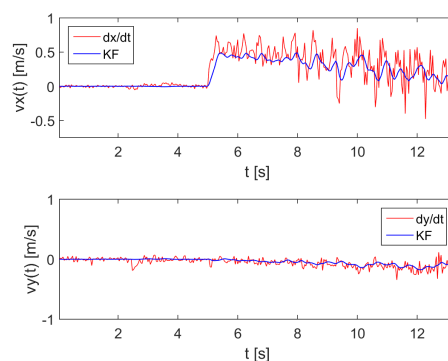
Os resultados obtidos com este estimador foram muito similares aos obtidos com o filtro de Kalman com o vetor de estado referenciado no referencial local. Verificou-se igualmente que quanto maior a distância de observação, e quanto maior a velocidade de rotação do robô, maior era o ruído presente na estimativa de velocidade.

Ensaio 3 - Robô Parado e Bola em Movimento

Esta secção corresponde ao ensaio 3 apresentado na secção 6.4.5.3, sendo que o filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção. Em comparação com os resultados do filtro de Kalman no referencial local, evidenciam-se algumas diferenças na velocidade estimada.



(a) Referencial local



(b) Referencial global

Figura 6.65: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos num robô real (Q3) - velocidade da bola

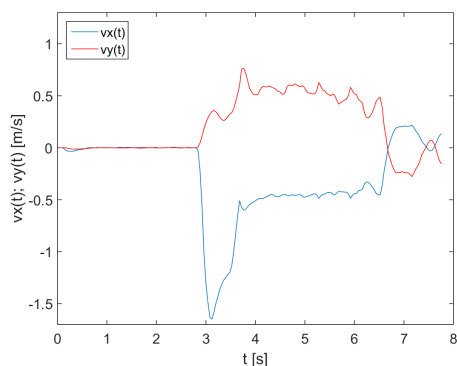
As figuras 6.65a e 6.65b contêm as componentes x e y da velocidade estimada da bola em função do tempo para a matriz de covariância Q3, nos referenciais local e global, respetivamente.

Comparando com os resultados obtidos no referencial local (figura 6.65), pode-se observar que se obteram resultados semelhantes, embora a dinâmica deste filtro tenha sido ligeiramente mais lenta. Note-se ainda que, neste filtro, foi necessário dar uma maior importância às medidas de modo a conseguir obter uma estimativa de velocidade semelhante à obtida na abordagem anterior. Isto deve-se principalmente ao facto de nesta abordagem se ter em conta a incerteza da localização do robô na matriz de covariância das medidas R , fazendo com que se dê uma maior importância ao modelo de transição de estado. Como este apresenta uma dinâmica aproximada da bola, é necessário ter uma matriz de covariância Q maior do que no filtro de Kalman no referencial local, de modo a dar a devida importância às medidas para se conseguir obter uma boa estimativa da velocidade da bola. Na figura 6.65a, é ainda visível que a velocidade estimada pelo filtro convergiu para a velocidade real da bola, obtendo-se um traçado mais suave do que a estimação a partir da derivada da posição em função do tempo.

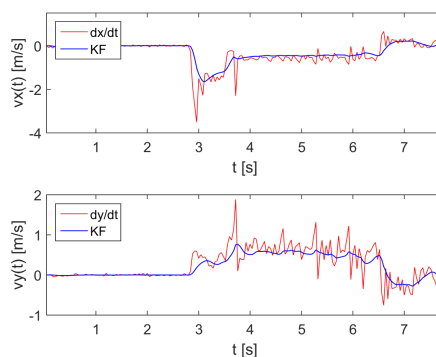
Ensaio 4 - Bola Contra Robô

Esta secção corresponde ao ensaio 4 apresentado na secção 6.4.5.3, sendo que o filtro de Kalman no referencial global teve os mesmos dados de entrada que o estimador apresentado na referida secção.

As figuras 6.66a e 6.66b contêm as componentes x e y da velocidade da bola em função do tempo, nos referenciais local e global, respetivamente, para o ensaio com a matriz de covariância $Q3$. Comparando com os resultados obtidos no ensaio com o filtro de Kalman local, verifica-se que obteve uma curva mais suave, porém não tão precisa e à custa de uma velocidade de convergência mais lenta. Em relação à estimação de velocidade através da derivada em função do tempo, verifica-se que os resultados obtidos com o filtro de Kalman foram mais suaves, embora com uma dinâmica ligeiramente lenta.



(a) Referencial local



(b) Referencial global

Figura 6.66: Ensaio 4 do filtro de Kalman com vetor de estado no referencial global, com dados obtidos num robô real ($Q3$) - velocidade da bola

6.5.2.4 Ensaios no *Software* de "Decisão" de cada robô

Em seguimento do sucesso dos ensaios realizados à implementação deste filtro de Kalman em Matlab, procedeu-se à implementação deste estimador no *software* de decisão da equipa 5DPO, tendo sido realizados ensaios em ambiente simulado (figuras A.1, A.2 e A.3) e real (figuras A.4, A.5 e A.6).

Note-se que os ensaios realizados nos robôs reais estão afetados pelo ruído presente nas medidas da odometria, assim como pelo erro de observação.

6.5.3 Inibição do Erro de Localização do Robô na Estimação da Velocidade da Bola

O filtro de Kalman dimensionado nesta secção tem o seu vetor de estado referenciado no referencial global. Como as observações são realizadas em cada agente, a sua conversão para o referencial global está dependente da localização do robô. Se o robô estiver bem localizado, a estimação da posição e velocidade ocorrem sem problemas de maior. No entanto, em ambiente real isto nem sempre acontece.

A localização dos agentes da equipa 5DPO resulta da fusão sensorial dos dados da odometria, visão e bússola digital. A equipa utiliza ainda um algoritmo de *perfect match* [68] para corrigir a localização dos robô com base na minimização dos erros de distância das medidas das linhas do terreno do jogo às mesmas.

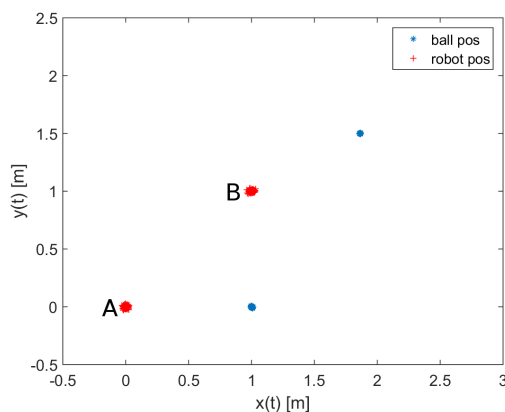


Figura 6.67: Vista de topo do cenário em que ocorre uma correção na localização de um agente, no referencial global

Poderá ser levantada a questão sobre o que acontecerá à dinâmica da bola se o robô receber uma correção da sua posição. Considere-se a situação de um agente parado a observar uma bola também ela parada. No instante $k = 6s$, o robô sofre uma correção da sua localização, passando do ponto A, com orientação de 0° , para o ponto B, com uma orientação de 30° (figura 6.67). Embora o robô continue a observar a bola na mesma posição em relação a si, ao converter a posição da

bola para o referencial global esta irá ser afetada pela translação e rotação que o referencial do robô sofreu. Se não forem tomadas precauções, como o cálculo da velocidade da bola tem em conta a posição global estimada para a bola, irá aparecer um pico na velocidade correspondente a esta variação repentina de posição. A figura 6.68a comprova a assunção anterior.

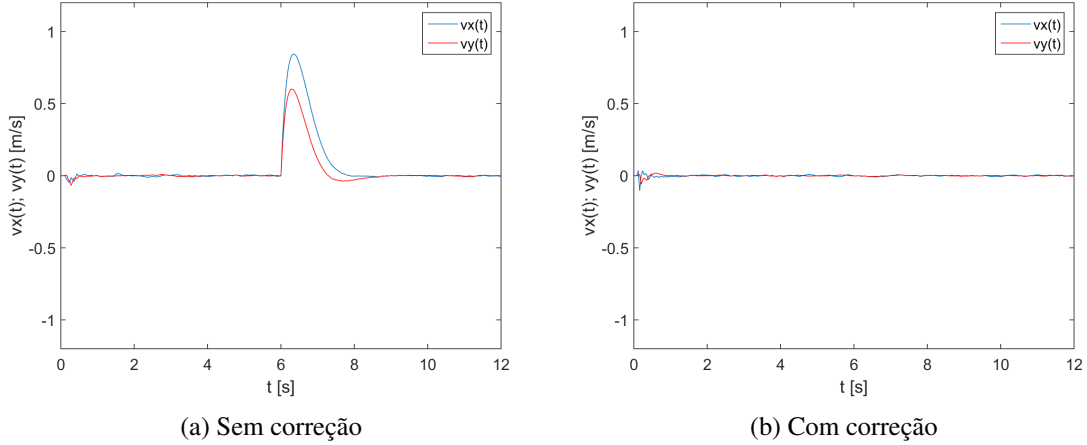


Figura 6.68: Velocidades da bola no referencial global no cenário de acontecer uma correção na localização do robô

Para evitar tal efeito, antes de ser realizada a fase de correção do filtro de Kalman da bola, começou-se a registrar a diferença entre a pose do robô (x_r, y_r, θ_r) , obtida através do filtro de Kalman estendido responsável pela fusão da localização do robô, e a pose $(x_{r_{odo}}, y_{r_{odo}}, \theta_{r_{odo}})$, obtida simplesmente pelos dados da odometria (equação (6.81)). Desta forma, $\Delta\theta$ e a matriz de translação T representam a correção efetuada na pose do robô pelos dados da visão. De seguida, aplicou-se à última estimativa do estado da bola \hat{X}_b^- , obtida na fase de predição do filtro de Kalman da bola, uma rotação $\Delta\theta$ e uma translação T , de modo a última bola estimada fosse afetada da mesma transformação que a pose do robô (equação (6.82)). Assim, quando se fosse calcular a inovação no filtro de Kalman, as bolas estimada e medida estariam ambas afetadas da mesma transformação. Note-se que embora se tenha utilizado a nomenclatura \hat{X}_b^- , este vetor corresponde às duas primeiras linhas do vetor de estado do filtro de Kalman da bola, sendo que as correções presentes na equação (6.82) eram aplicadas diretamente a estes elementos do vetor de estado.

$$T(k) = \begin{bmatrix} x_r(k) \\ y_r(k) \end{bmatrix} - \begin{bmatrix} x_{r_{odo}}(k) \\ y_{r_{odo}}(k) \end{bmatrix} \quad (6.81)$$

$$\Delta\theta(k) = \theta_r(k) - \theta_{r_{odo}}(k)$$

$$\begin{aligned} \hat{X}_b^-(k) &= R_{z, \Delta\theta(k)} \hat{X}_b^-(k) \\ \hat{X}_b^-(k) &= \hat{X}_b^-(k) + T(k) \end{aligned} \quad (6.82)$$

Contudo, esta correção deve ser avaliada com muito cuidado, pois, se efetuada em todos os ciclos, estará a injetar o ruído inerente à localização do robô diretamente no estado já estimado da

bola, aumentando o ruído neste. Por esse motivo, definiu-se que apenas se aplicaria as correções presentes em (6.82) no caso de a correção de posição do robô em cada direção se esta for superior a três vezes o desvio padrão da localização do robô nessa direção.

Tal como se pode verificar na figura 6.68b, esta correção tem um efeito bastante positivo na dinâmica da velocidade da bola, tornando-a independente das correções que o agente sofre na sua localização.

6.6 Estimação de Velocidade Através de uma Regressão Linear

Para obter um meio de comparação com outros métodos mais heurísticos, decidiu-se utilizar uma abordagem para a estimação da velocidade da bola semelhante à apresentada pela equipa CAMBADA, em 2010 [37, 43].

Esta abordagem parte do princípio de que a velocidade da bola é dada pela derivada da posição em ordem ao tempo. Denominando $B(k)$ como a posição da bola no instante k e considerando um sistema discreto amostrado a uma taxa ΔT , uma aproximação numérica para a velocidade da bola será:

$$v_b(k) = \frac{B(k) - B(k-1)}{\Delta T} \quad (6.83)$$

Contudo, este estimador está muito sujeito a ruído, tornando a estimativa de velocidade pouco precisa.

Para superar esse efeito, o estimador proposto consiste em armazenar um conjunto de pontos num *buffer* e calcular a regressão linear dos mesmos recorrendo ao método dos mínimos quadrados. Como a velocidade pode ser vista como a reta tangente ao traçado da posição em cada instante de tempo, a utilização de uma regressão linear de um conjunto de pontos obtém um efeito semelhante à derivada, tendo a vantagem de, devido a utilizar um conjunto de pontos, as estimativas serem mais suaves, ou seja, com menos ruído do que no caso de se calcular a derivada usando apenas dois pontos.

A dinâmica desta estimativa da velocidade pode ser controlada através do número de pontos a utilizar na regressão linear, sendo que, considerando o mesmo período de amostragem, quanto maior o número de pontos utilizados, menor vai ser a dinâmica da estimativa bola, tornando-se mais difícil de detetar transições bruscas de velocidade.

Para contrariar esse efeito, a equipa da Universidade de Aveiro sugeriu a redução do número de pontos a introduzir no *buffer* utilizado para o cálculo da regressão linear nos instantes em que eram detetadas transições bruscas na velocidade da bola [37, 43]. Devido ao pouco tempo restante para a implementação desta funcionalidade, decidiu-se definir como constante o número de pontos a utilizar na regressão linear.

Considerando um conjunto das últimas N observações consecutivas obtidas num instante de tempo T_1 , sendo x_m, y_m os vetores compostos pelas coordenadas x e y das medições da bola, \bar{x}_m e \bar{y}_m

o valor médio desses vetores, t o vetor que contém os instantes de tempo em que cada observação foi realizada e \bar{t} o valor médio desse vetor. Desta forma, a velocidade em T_1 é dada por [69]:

$$\begin{aligned} v_x(T_1) &= \frac{\sum_{k=0}^N (t(k) - \bar{t})(x_m(k) - \bar{x}_m)}{\sum_{k=0}^N (t(k) - \bar{t})} \\ v_y(T_1) &= \frac{\sum_{k=0}^N (t(k) - \bar{t})(y_m(k) - \bar{y}_m)}{\sum_{k=0}^N (t(k) - \bar{t})} \end{aligned} \quad (6.84)$$

Sabendo que o ciclo de controlo do programa de visão da equipa 5DPO fornece observações da posição da bola a cada 40ms e, após várias experiências, definiu-se 7 como o número de pontos a utilizar no cálculo da regressão linear. Este valor foi escolhido de modo a se obter uma dinâmica suficientemente rápida para detetar mudanças de trajetória da bola e, ao mesmo tempo, o menor ruído possível. O *buffer* consistia numa fila, sendo armazenadas sempre as últimas 7 medidas disponíveis da posição da bola. No caso de ainda não existirem pelo menos 7 observações (por exemplo, quando o programa é inicializado), realiza-se a regressão linear com o número de observações registado até ao momento.

6.6.1 Comparação de Estimação de Velocidade Através do Filtro de Kalman com Regressão Linear

Para avaliar a eficácia do estimador de velocidade com regressão linear utilizando o método dos mínimos quadrados, submeteu-se o mesmo aos ensaios realizados pelos dois filtros de Kalman em 6.4.5 e 6.5.2. A filtragem das observações da bola foi realizada através do filtro de Kalman com o vetor de estado referenciado no referencial local, apresentado em 6.4. Por forma a obter-se um meio de comparação, serão também apresentadas as estimações de velocidade do filtro de Kalman. Nos ensaios 1 e 2 utilizou-se a matriz de covariância do erro do modelo de transição de estado Q1 (equação (6.51)), e para o ensaio 3 utilizou-se a matriz Q2 (equação (6.52)), por serem as que apresentaram melhores resultados nos ensaios anteriormente apresentados, respetivamente.

Os ensaios foram realizados em ambiente de simulação, sendo que as ações do agente eram controladas pelo *software* de decisão, e a dinâmica do robô e da bola simulados pelo motor de simulação de corpos rígidos ODE. Para tornar a simulação mais realista, foi adicionado, no simulador, ruído à observação da bola por parte dos agentes na mesma ordem de grandeza do ruído existente nos robôs reais. Este ruído tinha em conta a distância a que o robô se encontrava da bola, de acordo com as expressões (6.45), (6.46) e (6.47).

Ensaio 1 - Robô a Descrever uma Trajetória de um Quadrado e Bola Parada

Este ensaio corresponde à situação em que o robô descreve a trajetória de um quadrado de lado 3m, em torno de uma bola estática. O robô realiza o seu movimento com uma velocidade linear de 1m/s, exceto nos períodos de aceleração e desaceleração perto dos vértices do quadrado.

A figura 6.69a contém as posições estimada e medida da bola, no plano xOy do referencial local, e a figura 6.69b as componentes x e y da posição estimada da bola em função do tempo, no referencial local.

As figuras 6.71a e 6.71b contêm as componentes x e y da velocidade estimada da bola, em função do tempo, no referencial local, respectivamente obtidas pelo método da regressão linear e através do filtro de Kalman.

As figuras 6.71a e 6.71b contêm as componentes x e y da velocidade estimada da bola, em função do tempo, no referencial global, respectivamente obtidas pelo método da regressão linear e através do filtro de Kalman.

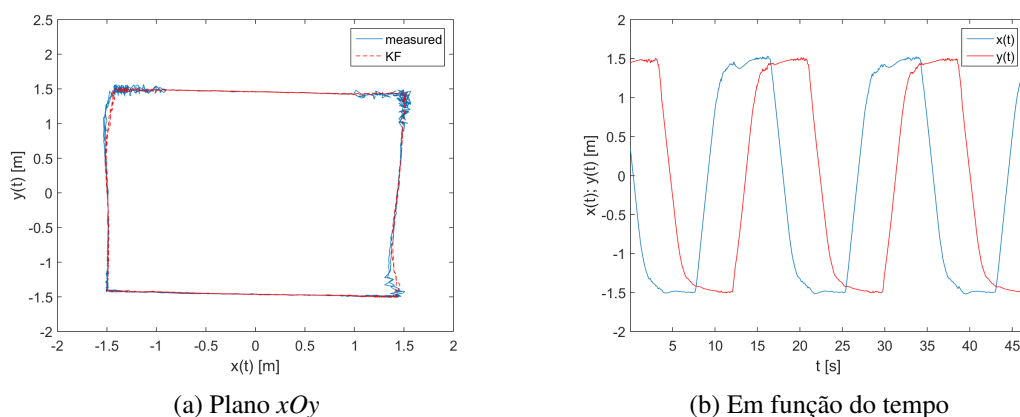


Figura 6.69: Ensaio 1 - Posições medida e estimada da bola, no referencial local

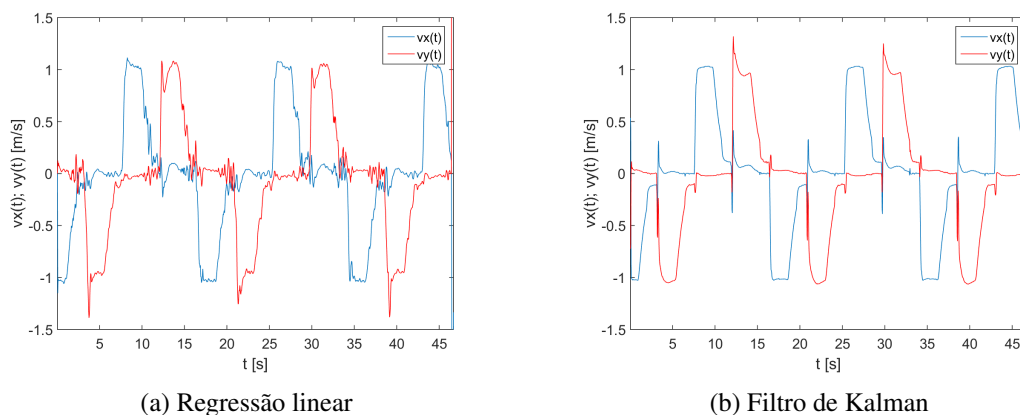


Figura 6.70: Ensaio 1 - Velocidade estimada da bola em função do tempo, no referencial local

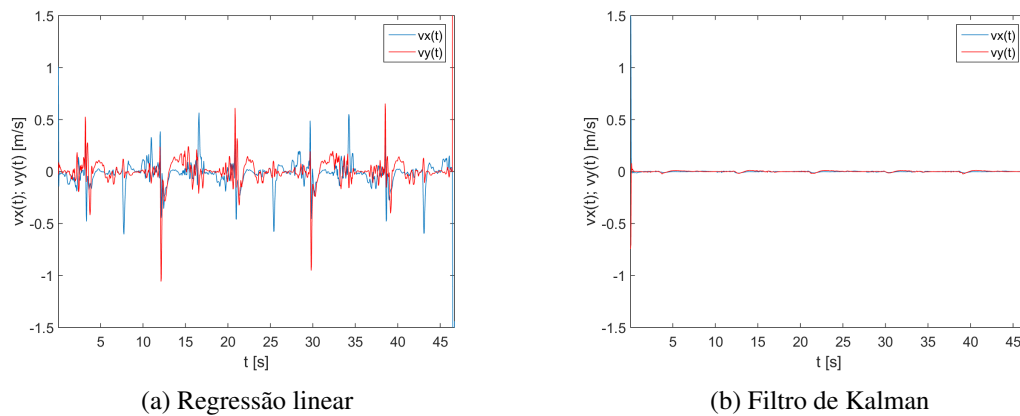


Figura 6.71: Ensaio 1 - Velocidade estimada da bola em função do tempo, no referencial global

Analisando estas figuras, é possível concluir que se obtiveram conseguiram obter melhores resultados na estimação de velocidade através do filtro de Kalman do que através da regressão linear. Note-se ainda que, através da utilização de um filtro de Kalman, obtém-se uma matriz de covariância que exprime a confiança que se pode ter na estimativa da velocidade.

Ensaio 2 - Robô a Rodar Sobre Si Mesmo e Bola Parada

Este ensaio corresponde à situação em que o robô roda sobre si mesmo, a uma velocidade angular de 0.9rad/s , à distância de 1m de uma bola estacionária, em relação ao referencial global.

A figura 6.72a contém as posições estimada e medida da bola, no plano xOy do referencial local, e a figura 6.72b apresenta as componentes x e y da posição estimada da bola em função do tempo, no referencial local.

As figuras 6.74a e 6.74b contêm as componentes x e y da velocidade estimada da bola, em função do tempo, no referencial local, respetivamente obtidas através do estimador dos mínimos quadrados e através do filtro de Kalman.

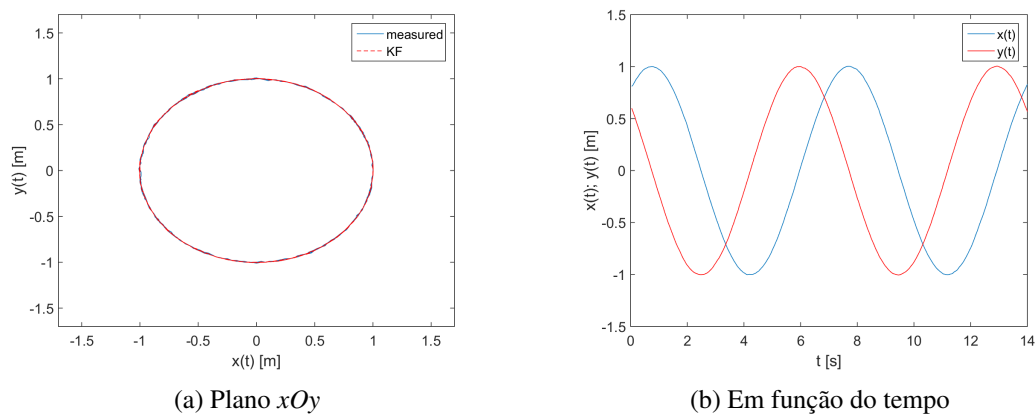
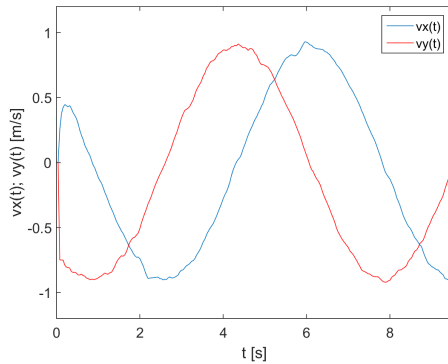
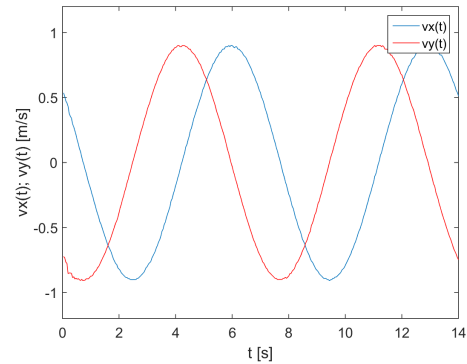


Figura 6.72: Ensaio 2 - Posições medida e estimada da bola, no referencial local

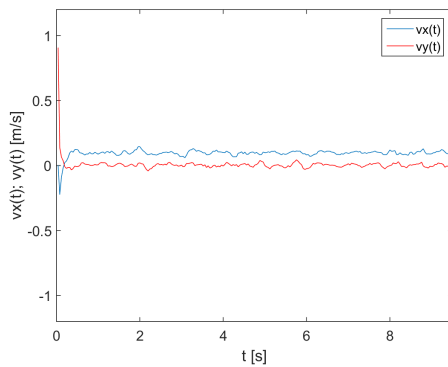


(a) Regressão linear

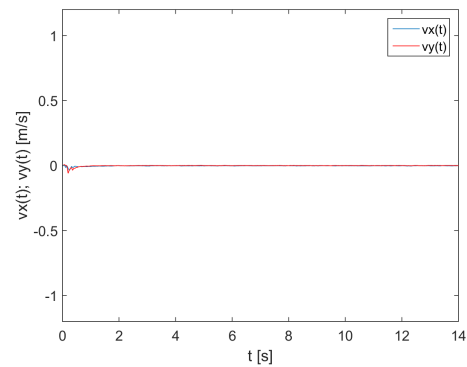


(b) Filtro de Kalman

Figura 6.73: Ensaio 2 - Velocidade estimada da bola em função do tempo, no referencial local



(a) Regressão linear



(b) Filtro de Kalman

Figura 6.74: Ensaio 2 - Velocidade estimada da bola em função do tempo, no referencial global

Analisando estas figuras, é visível que se obtiveram resultados similares, ligeiramente melhores na estimativa do filtro de Kalman.

Ensaio 3 - Robô Parado e Bola em Movimento

O terceiro ensaio corresponde à situação em que um robô parado observa uma bola em movimento a afastar-se de si, maioritariamente segundo a direção x do referencial global.

O gráfico da figura 6.75a contém as posições estimada e medida da bola, no plano xOy do referencial local, e a figura 6.75b apresenta as componentes x e y da posição estimada da bola em função do tempo, no referencial local.

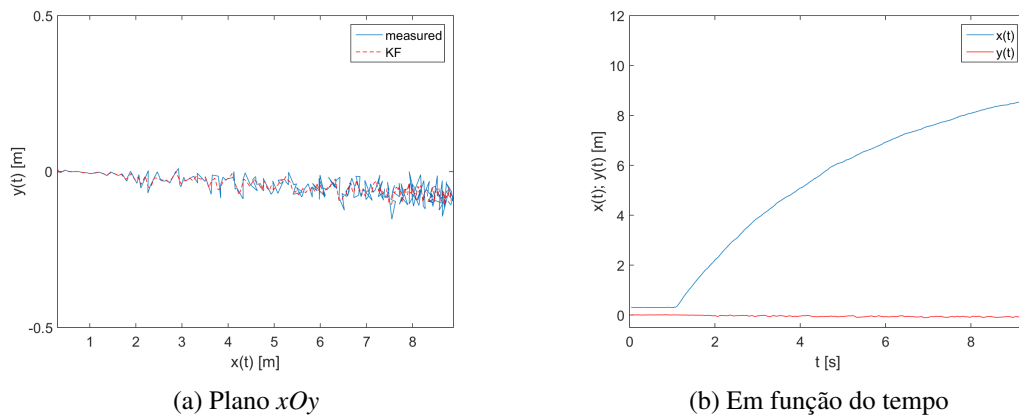


Figura 6.75: Ensaio 3 - Posições medida e estimada da bola, no referencial local

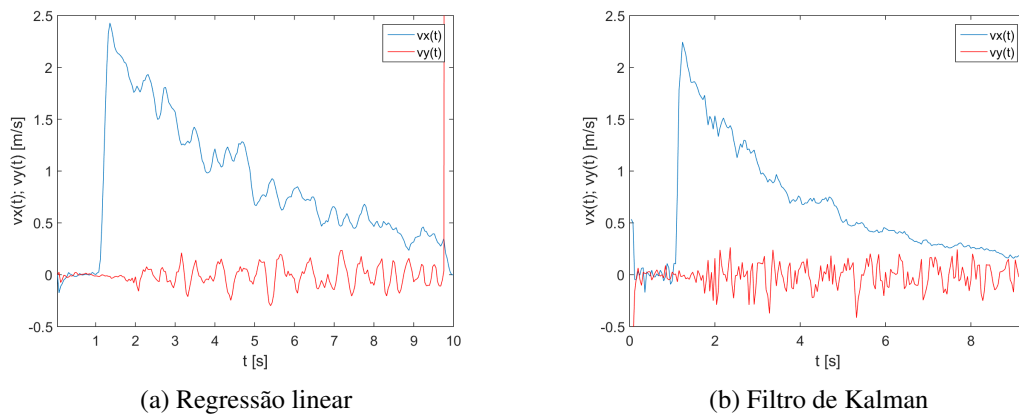


Figura 6.76: Ensaio 3 - Velocidade estimada da bola em função do tempo, no referencial local

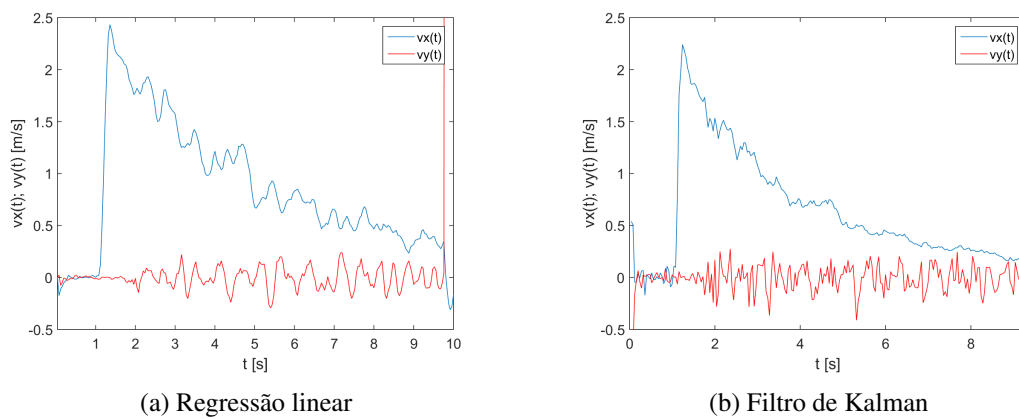


Figura 6.77: Ensaio 3 - Velocidade estimada da bola em função do tempo, no referencial global

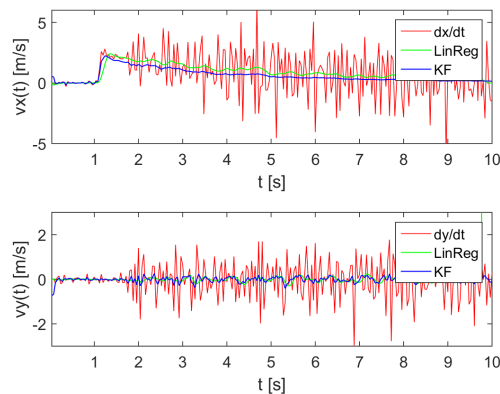


Figura 6.78: Comparação entre velocidades estimadas, no referencial local, por diferentes abordagens

As figuras 6.76a e 6.76b contêm as componentes x e y da velocidade estimada da bola, em função do tempo, no referencial local, respetivamente obtidas através da regressão linear e através do filtro de Kalman.

As figuras 6.77a e 6.77b contêm as componentes x e y da velocidade estimada da bola, em função do tempo, no referencial global, respetivamente obtidas através da regressão linear e através do filtro de Kalman.

A figura 6.78 contém os traçados das velocidades estimadas através da aproximação numérica da derivada em função do tempo, da regressão linear e do filtro de Kalman.

Numa primeira análise, pode-se concluir que se obtiveram resultados semelhantes. Contudo, observando pormenorizadamente a figura 6.78, verifica-se que a abordagem do filtro de Kalman tem uma dinâmica mais rápida do que a regressão linear. Por outro lado, a estimativa de velocidade obtida através da regressão linear encontra-se mais próximo do valor "real" da velocidade, considerando que, à falta de um sensor de *ground truth*, o valor estimado através da aproximação da derivada da posição em função do tempo aproxima da velocidade real da bola, à exceção do ruído.

6.7 Processo de Escolha de Bola no *Software* de Decisão

Até agora foram apresentados cenários em que o robô conseguia sempre realizar uma observação da bola, com boa qualidade. Todavia, numa situação de jogo real, isto nem sempre acontece, visto que a bola pode estar fora do alcance do sistema de visão ou escondida atrás de um robô. Outra situação que pode ocorrer é o facto de existirem pontos da imagem que o robô assuma como sendo uma bola, como por exemplo, uma reflexão do sol a bater no campo, uma peça de vestuário de cor parecida com a da bola, entre outros. Por estes motivos e, tal como em muitas outras aplicações, um agente deve dispor de mecanismos auxiliares que lhe permitam quantificar a

qualidade da sua estimativa de localização do alvo que está a seguir, assim como detetar e rejeitar observações falsas.

Os agentes da equipa 5DPO dispõem de alguns mecanismos que vão de encontro a esse fim. Em primeiro lugar, os *blobs* de possíveis bolas vindos do *software* de visão (*Ovis*) trazem associados uma qualidade, correspondente à relação do número de pixeis que o *blob* tem e que uma bola vista a essa distância deveria ter. Em segundo lugar, como é sabido que a bola só está em jogo dentro do terreno de jogo, são filtrados os *blobs* correspondentes a bolas localizadas fora de campo, no *software* de decisão (*Decision*). De seguida, é assumida como observação da bola o *blob* que, tendo sobrevivido até esta fase, tiver maior qualidade. O *Decision* utiliza um filtro de Kalman para estimar a posição e velocidade da bola. Em cada ciclo de controlo, é realizada uma previsão do estado da bola. Se a bola observada tiver qualidade positiva, a medida é utilizada na fase de correção do filtro de Kalman, sendo a bola estimada guardada num objeto à parte.

No filtro de Kalman, a confiança na estimativa do estado da bola é dada por uma matriz de covariância P , contendo as covariâncias da posição e velocidade nas direções x e y .

$$P = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{xv_x}^2 & \sigma_{xv_y}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{yv_x}^2 & \sigma_{yv_y}^2 \\ \sigma_{v_x x}^2 & \sigma_{v_x y}^2 & \sigma_{v_x v_x}^2 & \sigma_{v_x v_y}^2 \\ \sigma_{v_y x}^2 & \sigma_{v_y y}^2 & \sigma_{v_y v_x}^2 & \sigma_{v_y v_y}^2 \end{bmatrix} \quad (6.85)$$

Defina-se V como o produto das variâncias da posição da bola em x e y :

$$V = \sigma_{xx}^2 \sigma_{yy}^2 \quad (6.86)$$

Desta forma é possível definir a qualidade escalar da bola estimada pelo filtro de Kalman C , tal que:

$$C = \frac{1}{V} \quad (6.87)$$

Assim, quando maior for a incerteza na estimativa da posição da bola, maior será o valor das variâncias σ_{xx}^2 e σ_{yy}^2 , logo, maior será o valor de V e, consecutivamente, menor será C . Em contrapartida, quando menor for a incerteza na posição da bola estimada, menor será o valor das variâncias σ_{xx}^2 e σ_{yy}^2 , logo, menor será o valor de V e, consecutivamente, maior será C .

Quando o agente tem ligação ao treinador, recebe periodicamente uma estimativa de posição e velocidade da bola. Então, optou-se por estipular que se o valor de C fosse superior a um determinado valor C_{MAX} , o robô assumiria que não tem bola, passando a utilizar a bola oriunda do treinador. No caso de C ser inferior a C_{MAX} , o agente compara a sua bola estimada com a bola do treinador. Se se encontrarem próximas uma da outra, o robô assume a sua bola, senão, assume a do treinador. Se o robô não tiver ligação ao treinador assume sempre que a sua bola é a bola resultante da estimativa do filtro de Kalman.

6.8 Conclusões

Neste capítulo foram apresentadas duas abordagens de estimação de posição e velocidade de uma bola, observada num referencial móvel, a partir de um filtro de Kalman com o seu vetor de estado referenciado no referencial local, e um segundo filtro de Kalman, cujo vetor de estado se encontrava referenciado no referencial global. Foram realizados ensaios com dados gerados através de um *script* de Matlab, obtidos em ambiente de simulação e obtidos em robôs reais, tendo-se tentado que todos eles apresentassem condições mais de ruído e incerteza o mais próximo possível dos obtidos num robô real.

Ambos os filtros de Kalman apresentam vantagens e desvantagens. Analisando os resultados dos dois estimadores nas secções 6.4.5 e 6.5.2, pode-se concluir que em geral se obtiveram estimações de posição e velocidade passíveis de ser utilizadas pelos agentes numa situação de jogo. Todavia, as implementações contêm algumas diferenças que devem ser destacadas e analisadas.

O facto de o filtro de Kalman com o vetor de estado referenciado no referencial local não necessitar da localização do robô para realizar a estimação de estado, faz com que esta fique imune ao ruído de localização do mesmo. A estimativa de velocidade com o filtro no referencial local apresentou resultados mais exatos e com uma dinâmica mais rápida do que a abordagem no referencial global. Para além disso, mesmo no caso de o robô não ter localização, com o filtro no referencial local consegue obter na mesma uma estimativa da posição e velocidade da bola no referencial local, enquanto que com o filtro no referencial global não consegue.

Por outro lado, a dinâmica aparente da bola é diferente nas abordagens no referencial local e global. Por exemplo, caso de uma bola parada a ser observada por um referencial móvel, a bola no referencial local tem um movimento simétrico ao do robô, enquanto que no referencial global se encontra parada. Visto que ambos os filtros apresentam alguns erros de estimação devido à discretização das equações da dinâmica da bola, consegue-se obter melhores resultados num estimador onde a dinâmica da bola é mais suave do que mais variável.

Ambos os filtros se demonstraram capazes de recuperar de uma inicialização de velocidade errada no vetor de estado. Isto é um facto importante, visto que, enquanto que a posição pode ser inicializada com a primeira observação da câmara, o agente não tem nenhuma estimativa inicial de velocidade válida que lhe permita inicializar o filtro. Embora ambos os filtros tenham demonstrado uma boa dinâmica, deve-se destacar positivamente a resposta do filtro com vetor de estado no referencial local.

A matriz de covariância do ruído do modelo de transição de estado foi utilizada como parâmetro de afinação dos dois filtros de Kalman. Ao longo dos ensaios, teve-se a necessidade de se alterar as covariâncias desta matriz de modo a obter resultados mais precisos, nomeadamente quando a bola se encontrava em movimento, devido ao facto de o modelo utilizado na fase de transição de estado ser uma aproximação da dinâmica da bola, assim como o facto de esta ser complexa e imprevisível. Demonstrou-se que, ajustando a covariância do modelo de transição de estado, se conseguiu obter maior imunidade ao ruído das medidas, à custa de uma velocidade de convergência mais lenta. O valor ideal da matriz Q depende dos requisitos de cada situação.

Por vezes, pode-se querer um filtro mais rápido, mesmo que com mais ruído. Por outras vezes, pode ser preferível ter um filtro mais lento mas mais imune ao ruído. Desta forma, conclui-se que se deve utilizar uma matriz de covariância Q variável, dando uma maior importância às medidas em casos em que se sabe que a aceleração da bola será elevada, como por exemplo, durante um remate. Quando se pretende obter uma estimativa mais suave de bola, pode-se dar uma menor importância às medidas.

Uma curiosidade a ter em conta é o facto de, como no filtro de Kalman no referencial global o cálculo da matriz de covariância do erro da medida R contém a covariância da estimativa da pose do robô, verifica-se que, para iguais covariâncias das medidas da bola, ter-se-à uma matriz R superior no caso do filtro de Kalman no referencial global do que no referencial local. Consequentemente, a equação de transição de estado terá mais peso e suavizará as estimativas, tornando-as mais imunes ao ruído, à custa de uma dinâmica mais lenta. Isto pode ser uma possível explicação para o facto de se ter verificado uma dinâmica mais lenta no filtro de Kalman no referencial global.

A implementação do filtro de Kalman com vetor de estado referenciado no mundo para estimação de posição e velocidade da bola no *software* de decisão da equipa revelou um bom desempenho, tanto em ambiente de simulação como em testes em robôs reais.

Ainda no filtro de Kalman com o vetor de estado referenciado no referencial global, a correção realizada ao estado estimado da bola, no caso do robô receber uma correção de localização, demonstrou-se extremamente eficaz, eliminando o erro que iria aparecer na velocidade estimada da bola devido à mudança de posição da bola observada.

Para além da estimação de velocidade através de um filtro de Kalman, realizou-se ainda a estimação de velocidade através duma regressão linear, correspondendo a uma simplificação da abordagem apresentado pela equipa CAMBADA [37, 43]. Pode-se verificar que, em geral, se obtiveram resultados muito semelhantes, havendo ligeiras vantagens para a abordagem do filtro de Kalman, nomeadamente, estimações mais suaves quando a bola está parada, e uma convergência mais rápida para a velocidade da bola no caso de esta se encontrar em movimento. Contudo, note-se que, neste último caso, a estimativa da velocidade obtida através da regressão linear apresentou um valor ligeiramente mais exato que a do filtro de Kalman. Note-se ainda que, utilizando um filtro de Kalman, para além da estimativa da posição e velocidade da bola, obtém-se uma matriz de covariância que representa a confiança que se pode ter nessa estimativa.

Em suma, conseguiu-se atingir o resultado pretendido, nomeadamente a estimação de posição e velocidade de um alvo. Das diferentes implementações só houve tempo para integrar no *software* de decisão a abordagem do filtro de Kalman cujo vetor de estado se encontra referenciado no referencial global. Contudo, tendo por base os resultados obtidos na implementação em Matlab, acredita-se que a utilização do filtro de Kalman com o vetor de estado referenciado no referencial local no *Decision* terá resultados possivelmente melhores que a abordagem até agora implementada, principalmente quando forem resolvidos os problema do ruído de odometria descobertos nestes ensaios.

Capítulo 7

Treinador de equipa de Futebol Robótico

O desenvolvimento tecnológico que tem decorrido nos últimos anos tem proporcionado múltiplas soluções robóticas para as mais diversas áreas de aplicação. A utilização de múltiplos robôs tem sido igualmente explorada, proporcionando um serviço com uma área de cobertura maior e com menores custos. Desta forma, a cooperação entre agentes assume um papel bastante relevante na maximização de eficiência de uma equipa de robôs móveis.

O futebol robótico levanta uma série de desafios a nível da robótica, principalmente a nível de cooperação, visto que o futebol é um jogo de equipa, onde o bom desempenho individual dos agentes não assegura uma boa exibição da equipa. Durante um jogo, tem de existir comunicação e consenso entre os agentes acerca das ações a desempenhar por cada um, garantindo um bom posicionamento ao longo do campo, bem como adaptabilidade face às diferentes situações de jogo. A *Middle Size League* proporciona um ambiente ideal para desenvolvimento, implementação e testes de novas técnicas de cooperação. Nesta liga, cinco agentes autónomos agem em conjunto e são submetidos aos mais diversos desafios. Estes devem ser capazes de perceber o mundo que os rodeia, identificar um alvo (a bola), e posicionar-se estrategicamente no campo de modo a atingirem um objetivo comum, seja ele defender a sua baliza, marcar um golo, passar tempo, entre outros.

Nesse sentido, a presente dissertação propõe a utilização de um supervisor que garanta a correta coordenação dos agentes ativos, decidindo qual o melhor papel a desempenhar por cada um, e garantindo que estes se comportam de acordo com a situação atual do jogo.

Este capítulo apresenta o supervisor responsável pela cooperação dos agentes da equipa de futebol robótico MSL 5DPO. Começará por ser apresentada a interface do treinador, e serão explicadas as suas principais funcionalidades. De seguida, serão analisadas a arquitetura da aplicação, o seu ciclo principal de controlo, as formações utilizadas nas diversas situações de jogo e será explicado o sistema de atribuição de papéis a cada jogador. Por fim, será abordado o novo mecanismo de comunicações adotado pela equipa, assim como outras funcionalidades implementadas neste supervisor.

7.1 Novo Treinador da Equipa 5DPO

A figura 7.1 contém uma imagem da interface do novo treinador da equipa 5DPO desenvolvido durante esta dissertação.



Figura 7.1: Novo treinador da equipa 5DPO

Este supervisor foi desenvolvido segundo a arquitetura de cooperação apresentada em 4.4, voltando a reutilizar o conceito de tática de equipa. O seu principal objetivo é garantir a coordenação e cooperação dos agentes da equipa em campo, decidindo qual o melhor papel a desempenhar por cada jogador para cada situação de jogo. Como é executado na *Base Station* da equipa, o treinador é responsável por interpretar os comandos enviados pela *RefereeBox* e adaptar o comportamento dos agentes em função dos mesmos.

Este *software* tem também um importante papel na monitorização do seu estado interno e no dos restantes agentes da equipa. Na parte superior da janela, é possível visualizar a tática, jogada, estado do árbitro, da bola e do jogo em cada instante do tempo. Do lado direito, é possível obter informações do estado dos agentes, como os seus papéis, tarefas, voltagem das baterias, e tempo desde a última mensagem recebida. Cada agente pode ter um de três estados associados: *active*, *ready* ou *disconnected*. O primeiro indica que as comunicações com um robô foram estabelecidas com sucesso, este encontra-se pronto e está a ser integrado pelo treinador na cooperação com a equipa. O segundo tem um significado semelhante ao primeiro, mas indica que o supervisor não está a incluir o agente na cooperação da equipa. O terceiro indica que um robô ou não está a comunicar com o treinador, ou não se encontra pronto para cooperar. No caso do agente não ter localização, o supervisor altera o estado do jogador para *No Loc*, permitindo que os responsáveis pela equipa decidam entre a continuidade ou remoção do agente do campo.

Na parte central da janela da aplicação é apresentado um mapa 2D que contém uma representação do estado do mundo. Este contém a posição e orientação de todos os agentes da equipa, assim como os seus papéis e estimativas de posição da bola (com respetivo grau de qualidade/confiança). É ainda representada a vermelho a estimativa de bola do treinador.

Para facilitar o *debug* de novas jogadas e formações, o treinador permite forçar diretamente as variáveis mais importantes do seu estado interno (i.e., tática, jogada, estado do árbitro, posse e conhecimento de bola, etc), assim como o papel atribuído individualmente a cada agente. Este programa possibilita ainda a simulação de uma *RefereeBox* localmente, facilitando assim a visualização do cenário do jogo durante a realização de testes.

7.2 Arquitetura da Aplicação

Analisando macroscopicamente o sistema, é possível identificar um conjunto de objetos com características próprias bem definidas. Perante a complexidade do problema proposto e, para garantir uma modularidade do sistema, decidiu-se aplicar uma arquitetura orientada a objetos. Para além disso, a possibilidade de encapsular certos atributos ajuda a garantir uma boa estrutura do código a curto e longo prazo.

No que concerne a linguagem de programação, decidiu-se utilizar a linguagem Object Pascal, devido à sua capacidade de cumprir os requisitos temporais de uma aplicação em tempo real, e à possibilidade de desenvolvimento de aplicações orientadas a objetos com interface gráfica de uma forma rápida e com uma baixa complexidade. Para além disso, uma grande parte do *software* da equipa está desenvolvido nesta linguagem, mantendo-se uniformidade no desenvolvimento das aplicações, e exigindo menores esforços numa fase posterior de manutenção de código. Como ambiente de desenvolvimento foi utilizado o IDE Lazarus ¹, devido à facilidade de desenvolvimento de aplicações com interface gráfica, e à já existência de componentes específicos da equipa 5DPO para este IDE.

A arquitetura principal do supervisor desenvolvido pode ser observada na figura 7.2. A classe principal (TCoach) contém um conjunto de objetos de forma a conseguir construir uma boa representação do mundo. Nesse sentido, é composta por um objeto TField, que contém as dimensões do campo; um objeto TRefereeBox, que contém a informação correspondente ao árbitro, tal como o estado e comando recebido, sendo responsável pela atualização do mesmo; um objeto TGameState, que é responsável pela gestão do estado do mundo, tal como o número de golos de cada equipa, tempo de jogo decorrido e estado do jogo; sete objetos TSdpoPacket e TSdpoZMQ, que são utilizados para as comunicações com o *software* de decisão presente em cada robô; um objeto TCoachBall, que contém a informação referente à gestão da bola do treinador e, finalmente, um objeto TTeam, que contém toda a informação referente à equipa. Esta contém uma tática (TTactic), uma jogada (TPlay), uma cor (ETeamColor) e seis objetos correspondentes aos agentes da equipa (TPlayer). A classe TPlayer contém a informação dos jogadores, como o seu estado (ativo/desativo), posição, velocidade, estado da bateria e do sensor de bola, papel (TRole) e tarefa

¹<https://www.lazarus-ide.org/>

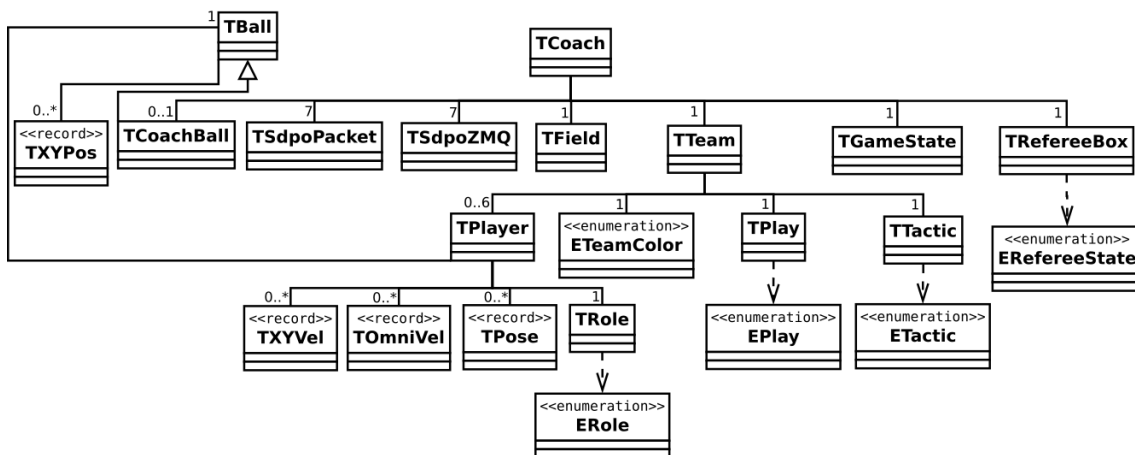


Figura 7.2: Diagrama UML de classes do treinador

que está a executar. Uma descrição mais pormenorizada acerca das principais classes e enumerações desta aplicação encontra-se disponível no anexo B.1.

7.3 WorldState

A construção e conhecimento do estado do mundo é um dos principais requisitos para se obter uma boa cooperação. De facto, para que o supervisor consiga determinar qual o melhor jogador para desempenhar um determinado papel necessita de ter conhecimento da posição dos mesmos e da bola. Para além disso, o estado da bola também influencia a escolha de jogada para cada situação de jogo.

Na equipa 5DPO, o treinador constrói o seu próprio estado do mundo com base na informação percecionada pelos agentes da equipa. A receção de dados é garantida por seis *threads* que correm paralelamente e estão à escuta e de mensagens vindas dos diferentes agentes. A informação partilhada é constituída pela posição, velocidade, tarefa em execução, tensão das baterias, estado do sensor de bola e da localização (se o robô se encontra localizado ou não localizado), assim como a posição, velocidade e confiança da bola que observaram.

Como a observação da bola recebida dos diversos agentes pode assumir valores, é realizada uma fusão da mesma, de modo a se obter uma estimativa mais precisa e, consequentemente, escolhas mais apropriadas na atribuição dos papéis. Esta fusão será abordada mais detalhadamente no capítulo 8.

7.4 Ciclo Principal

Nesta secção será analisado o ciclo principal do supervisor desenvolvido nesta dissertação (figura 7.3).

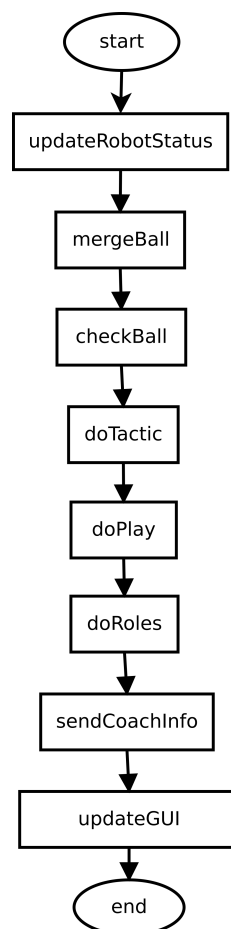


Figura 7.3: Ciclo principal de controle

Numa primeira fase, a partir da informação recebida de cada agente, é atualizado o estado dos agentes (*active*, *ready* ou *disconnected*), assim como o número de agentes ativos em campo. A restante informação dos agentes (localização, velocidades, bola e estado dos sensores) é atualizada assincronamente por uma função de *callback* evocada pela *thread* responsável pela comunicação com cada agente. A função de *callback* corre sincronamente com o programa principal, evitando problemas de acessos aos mesmos dados por dois processos concorrentes.

De seguida, é realizada uma fusão do estado da bola. Começam por ser filtradas as bolas fora de campo e, de seguida, é escolhida a bola do agente que tiver maior qualidade, atualizando o estado da mesma. O capítulo 8 aborda outras técnicas alternativas para a fusão sensorial da bola.

Seguidamente, com base no *worldstate* e no estado da *RefereeBox*, são executados os algoritmos de atualização da tática e jogada da equipa descritos em 7.4.2 e 7.4.3. A seguir, é atualizada a máquina de estados correspondente aos papéis para a jogada utilizada, é escolhido qual o robô mais adequado para ocupar uma determinada *slot*, e são atribuídos aos agentes os papéis correspondentes a cada *slot*.

De seguida, são enviadas aos jogadores informações relativamente às posições, papéis e estado de todos os agentes, assim como a estimativa de posição e velocidade da bola que o supervisor

realizou.

Por fim, é atualizada a interface gráfica da aplicação.

7.4.1 Atualização do Estado da *RefereeBox*

Na *Middle Size League* existem regras semelhantes às aplicadas nos jogos de futebol tradicional que condicionam e/ou geram situações de jogo. As decisões do árbitro são enviadas para as equipas através de comandos fornecidos por um programa denominado *RefereeBox*. Na equipa 5DPO, o treinador é o elemento responsável por traduzir os comandos da *RefereeBox* em alterações no comportamento da equipa. Desta forma, definiram-se um conjunto de estados para o árbitro, sendo estes utilizados posteriormente na escolha da tática e jogada mais adequada para a equipa em cada momento. Estes estados refletem as instruções dadas pelo árbitro, tais como início ou pausa de jogo, *parking*, ou situações específicas para cada equipa como penalti, falta, livre, canto, pontapé de baliza, lançamento ou golo, por exemplo. Tal como é visível na figura 7.4, a atualização do estado da *RefereeBox* é efetuada a partir dos comandos recebidos do mesmo e da cor atribuída à equipa. A atualização do estado do árbitro é realizada quando se verifica a receção de um novo comando.

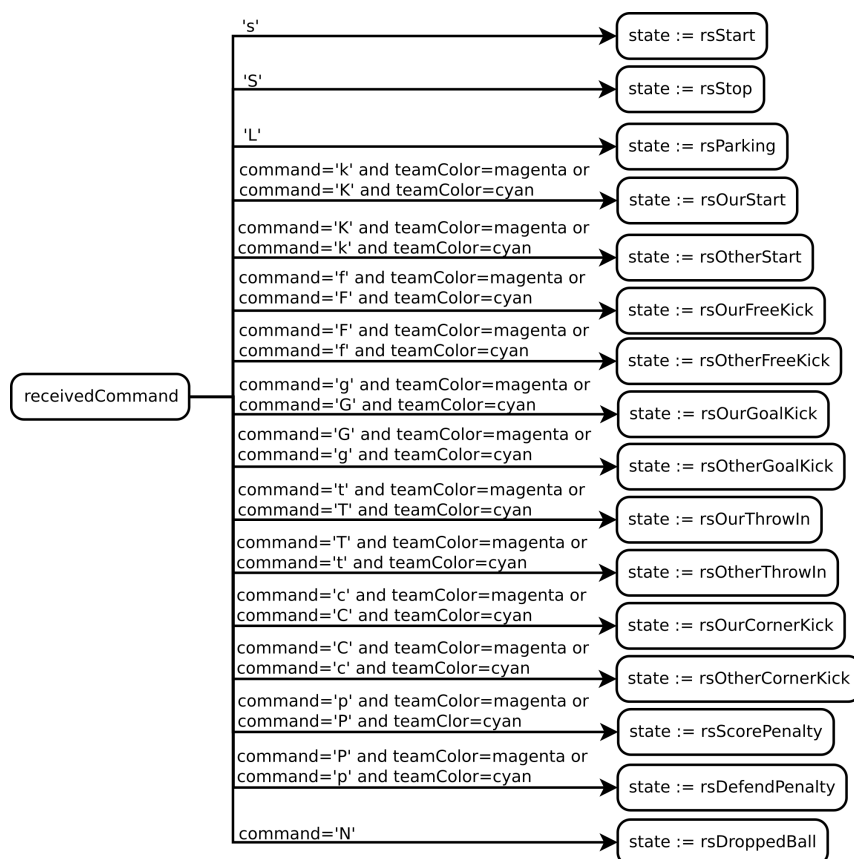


Figura 7.4: Diagrama de atualização do estado da *Referee Box*

7.4.2 Atualização da Tática

A atualização de tática tem por base o estado do árbitro, do jogo e do mundo, em cada instante de tempo. Foram definidas seis táticas, que pretendem representar a intenção da equipa atacar fortemente, defender fortemente ou outros níveis intermédios. Assim, em cada ciclo do programa principal é avaliado o estado do árbitro e, mediante o diferencial de golos, número de agentes em campo e tempo restante de jogo é escolhida a melhor tática para a equipa. A árvore de decisão da tática de equipa encontra-se representada no anexo [B.2](#).

7.4.3 Atualização da Jogada

A cada ciclo do programa principal, é corrida a máquina de estados responsável pela atualização da jogada da equipa. As jogadas são parametrizadas, contendo um estado (ativas ou não), mediante o jogo se encontra a decorrer ou parado, e permitem ter conhecimento da jogada anterior, futura e tempo atual da jogada. Atualmente, a equipa conta com sete jogadas distintas, tal como é visível na figura [7.5](#). As jogadas *playOurStart*, *playBarrier*, *playDroppedBall*, *playScorePenalty* e *playDefendPenalty* podem ser vistas como um super estado, e representam uma situação específica no jogo, vulgarmente denominadas como *set pieces*. Desta forma, quando o estado do árbitro indica que foi marcado um lançamento, falta, pontapé de baliza, etc para qualquer uma das equipas, o treinador ajusta a jogada de acordo com a mesma. Se a localização da bola for conhecida, a jogada é atualizada com o estado inativo (visto que normalmente estas transições ocorrem após o árbitro ter parado o jogo). Porém, se a localização da bola não for conhecida, a jogada utilizada mantém-se na *playSearchBall*, sendo atualizado o campo *nextPlay* para a jogada que seria escolhida. Um processo corrido em *background* avalia o conhecimento da localização da bola e, caso se verifique uma transição de um estado de não conhecimento para um estado de conhecimento, é atualizada a jogada utilizada para aquela presente no campo *nextPlay* da jogada atual. Quando o árbitro dá ordem de início da jogada (através do comando *Start*), esta passa para o estado ativo e, após sete segundos de atividade, passa automaticamente para a jogada *playNormal*. Esta transição recai no facto de poder haver uma modificação no estado do mundo que torne a distribuição de papéis escolhida para uma determinada jogada desadequada para a situação atual. No caso do árbitro enviar um sinal de paragem do jogo (comando *Stop*), a jogada passará para *playSearchBall* até ser recebido um novo comando referente à próxima situação de jogo. Para além disso, se a localização da bola deixar de ser conhecida em qualquer instante de tempo, a jogada passará igualmente para *playSearchBall*, sendo atualizado o campo *nextPlay* para a jogada anterior a esta transição. Assim que a localização volte a ser conhecida e detetada pela máquina de estados *checkBall*, a jogada regressará à sua jogada anterior (guardada no campo *nextPlay*).

7.4.4 Papéis, Tarefas e Ações

Por sua vez, os papéis, tarefas e ações estão localizados no *software* de decisão. Os papéis definem o comportamento dos agentes em campo, definindo uma sequência de tarefas a executar pelo mesmo. A execução da tarefa pode estar associada a um objetivo, que pode ir desde um bom

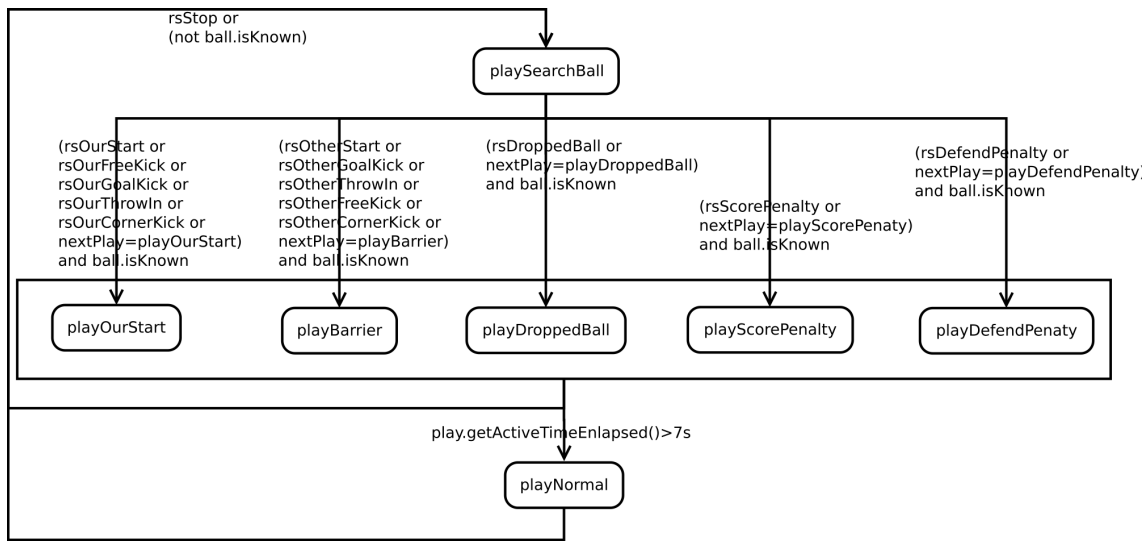


Figura 7.5: Diagrama de atualização da jogada da equipa

posicionamento em campo, até à interação direta ou indireta a bola. Para garantir a correta cooperação dos agentes em campo, a atribuição de papéis aos mesmos é realizada pelo treinador. A área de ação dos papéis está definida internamente na sua implementação, podendo estar ou não restrita a uma determinada zona do campo. Desta forma, torna-se possível definir um conjunto de formações que garantam um bom posicionamento ao longo do campo, assim como um comportamento adequado perante a situação de jogo em cada instante do jogo.

7.5 Formações e Atualização de Papéis

Durante um jogo, o principal objetivo do treinador é garantir a cooperação dos agentes da equipa em campo. A mentalidade da equipa pode ser aproximadamente caracterizada e definida pela jogada ativa em cada momento. Por exemplo, durante um pontapé livre a favor da equipa adversária, é utilizada a jogada *playBarrier*, e os robôs devem tomar um comportamento maioritariamente defensivo. Por outro lado, num pontapé livre a favor da equipa 5DPO, é ativada a jogada *playOurStart*, e os robôs devem ser capazes de executar uma movimentação ofensiva.

Consequentemente, propôs-se que a cooperação da equipa 5DPO fosse baseada em formações escolhidas em conformidade com a jogada atual. Visto que na equipa, o cálculo da posição de destino está definido internamente na implementação de cada papel (em cada robô), o treinador começa por definir um conjunto de papéis de modo a que os robôs ocupem certas posições estratégicas em campo, descrevendo, ao mesmo tempo, um comportamento adequado para a situação de jogo em questão. Cada equipa pode conter no máximo cinco jogadores em campo, porém, devido a avarias ou punições por parte do árbitro, alguns agentes podem ser obrigados a sair de campo, deixando a equipa com um número inferior de jogadores. Para suportar estas oscilações em campo, definiram-se cinco *slots* e atribuí-se um papel a cada *slot* de ordem decrescente de

importâncias, i.e., o papel mais importante é atribuído ao *slot* 0, o segundo mais importante ao *slot* 1, e assim sucessivamente. Posteriormente, é definido qual o robô mais adequado a ocupar um determinado *slot*, atribuindo-se um *slot* e, conseqüentemente, um papel ao mesmo. Por forma a tornar o comportamento dos agentes mais adequado à situação de jogo, em vez de se definir um conjunto de papéis estáticos para cada *slot*, define-se uma máquina de estados que vai transitando de papel em papel, em conformidade com o desenrolar da jogada.

Situação de *Set Pieces*

Suponha-se uma situação de livre para a equipa adversária (figura 7.6). Perante esse cenário, a equipa assume um comportamento defensivo, pelo que se estabeleceu que o papel mais importante é o do guarda-redes, seguindo-se um agente colocado a fazer barreira à distância mínima possível da bola, um defesa colocado no exterior da grande área, na linha da bola com a baliza e, por fim, dois defesas a cobrir possíveis passes para as zonas de canto. Os papéis correspondentes a estes comportamentos são dados respetivamente por *roleKeeper*, *roleBarrierHighPressure*, *roleLastDefender*, *roleBarrierHalfLeft* e *roleBarrierHalfRight*. Como em qualquer *set piece* é obrigatória a execução de um passe, quando a bola iniciar movimento é declarado que esta entrou em jogo, fazendo com que os dois defesas que estão na extremidade do campo assumam o papel *roleClearBall*, cujo objetivo é ir tentar aliviar a bola, e o agente que está a executar a barreira mais avançada passa ao papel *roleAttackerKicker*, indo de encontro à bola com intenções atacantes (que também se revelam extremamente eficazes defensivamente na execução de blocos e/ou pressão alta). Estas transições também se podem verificar no caso de passarem sete segundos e a equipa atacante não ter tomado iniciativa de ataque. Quando o atacante da equipa 5DPO alcançar a bola, um dos agentes com o papel *roleClearBall* passa a auxiliar o atacante, enquanto o segundo ocupa uma posição central no campo.

Situação de Jogo Normal

Durante uma jogada de *playNormal*, a determinação e identificação do cenário de jogo não é tão trivial como numa situação de *Set Piece*. Como tal, a decisão acerca da distribuição de papéis deve ter em consideração outros fatores que permitam adequar o comportamento dos agentes à situação atual de jogo. Para isso, utilizou-se a tática escolhida pelo supervisor como principal elemento diferenciador, permitindo reconhecer situações em que se adequa mais uma formação defensiva ou atacante.

A figura 7.7 contém as formações e máquinas de estados definidas para cada cenário da jogada *playNormal*. Se a equipa tiver uma tática defensiva (*tacticDefend* ou *tacticDefendResult*), é definida uma formação defensiva, composta por um guarda-redes (*roleKeeper*), um jogador que faça pressão alta e tente aliviar a bola (*roleClearBall*), e três agentes que se coloquem na linha da bola com a baliza em diversas zonas do campo (*roleLastDefender*, *roleSecondLineDefender* e *roleMidfield*). Por outro lado, se equipa não tiver uma tática defensiva, dois defesas assumem papéis atacantes. Como as regras desta liga exigem que seja realizado no mínimo um passe antes

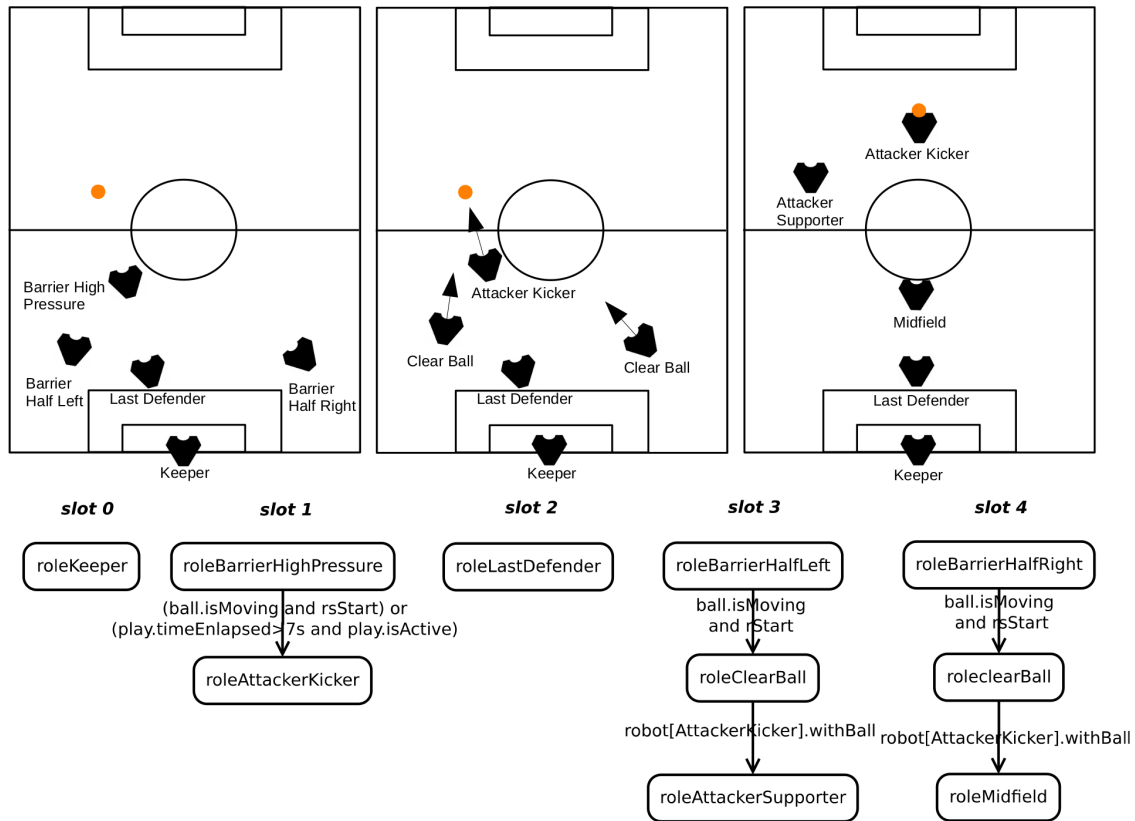
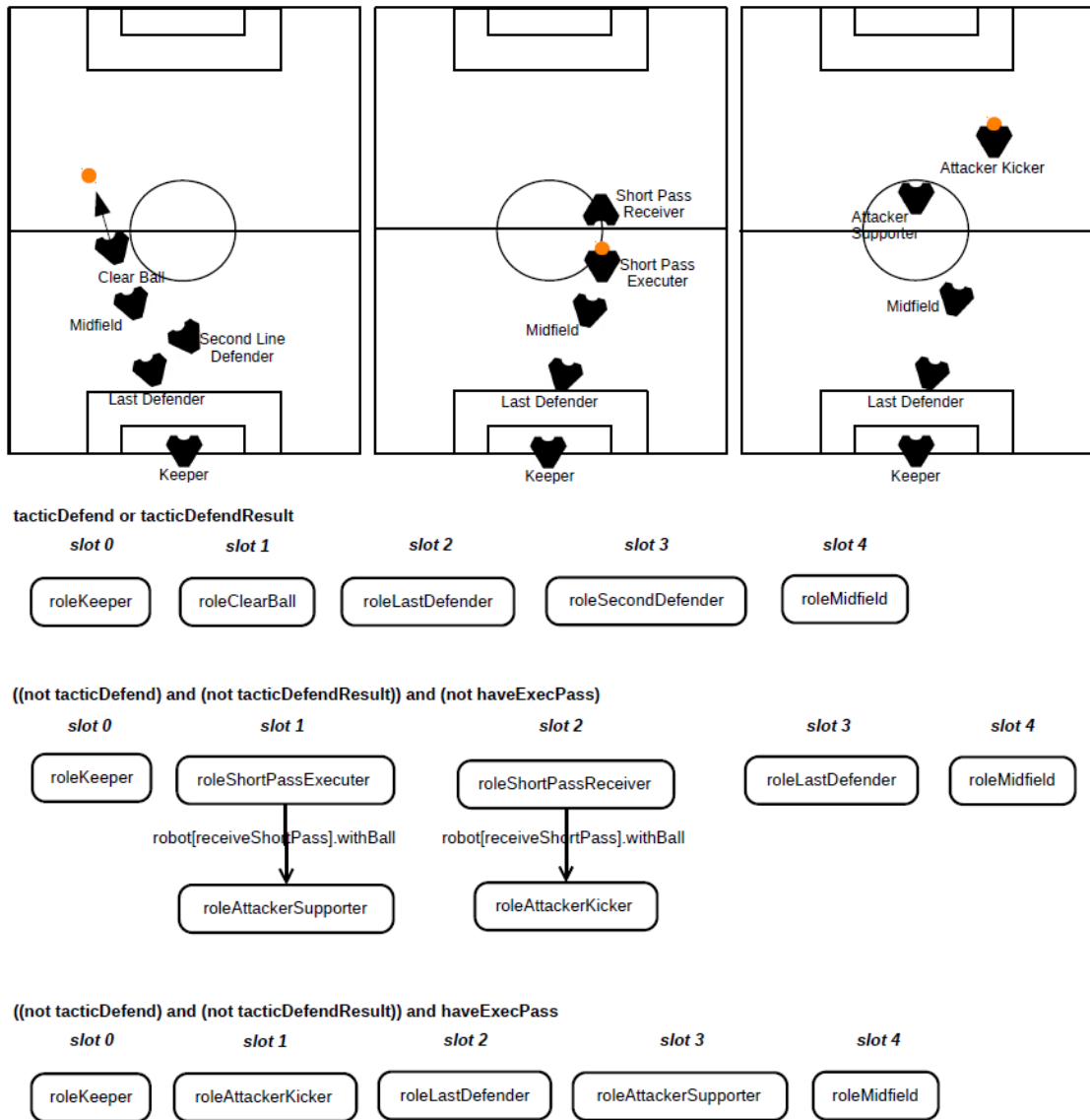


Figura 7.6: Formações e máquina de estados para a jogada *playBarrier*

da concretização de um golo, é realizada uma verificação se a equipa já efetuou um passe. Se ainda não tiver efetuado, um agente com o papel *roleShortPassExecuter* tentará passar a bola a um companheiro de equipa, preferencialmente um segundo jogador com o papel *roleShortPassReceiver*. Quando este tiver rececionado a bola, passará a atacante principal (*roleAttacker Kicker*), e será auxiliado pelo agente que realizou o passe (*roleAttackerSupporter*). No caso do passe ser interetado pela equipa adversária, a tática mudará para *tacticDefend* ou *tacticDefendResult* e, consequentemente, a equipa passará para a formação defensiva anteriormente descrita. A situação descrita pela terceira máquina de estados representa a formação de ataque da equipa, e verifica-se quando existe uma transição para a jogada *playNormal* em que já se tenha efetuado um passe na jogada anterior. As formações para as restantes jogadas adotam uma abordagem semelhante às apresentadas anteriormente, podendo ser consultadas no anexo B.3.

7.6 Atribuição Dinâmica de Roles

Em aplicações multi-agente inseridas em ambientes altamente dinâmicos não é suficiente garantir que a equipa consiga cooperar entre si. Embora haja vários agentes disponíveis para executar uma determinada tarefa, o custo de execução não é igual para todos, quer seja porque um robô se

Figura 7.7: Formações e máquina de estados para a jogada *playNormal*

encontra mais perto do ponto destino, porque demora menos tempo a terminar a tarefa ou até porque é o único que tem autonomia de bateria suficiente que lhe permita acabar a tarefa. Desta forma, para além de assegurar que os agentes têm os seus papéis bem definidos, é crucial que o supervisor garanta que estes se encontram corretamente distribuídos de modo a assegurar o desempenho ótimo da equipa.

O futebol robótico levanta vários desafios neste sentido. Durante um jogo, há múltiplas variáveis que mudam de estado de um momento para outro. O objetivo da equipa tão depressa pode ser atacar a baliza adversária, como defender a própria, e a própria bola pode mudar a sua localização muito rapidamente. Outro fator a ter em conta é a possibilidade de haver variações no número de agentes operacionais, devido a punições do árbitro ou, no caso de ligas que consistem

em implementações em robôs reais como a MSL, devido a avarias de *hardware*.

O treinador desenvolvido nesta dissertação procura encontrar qual o melhor agente para os papéis estabelecidos pelas jogadas através do cálculo do custo que os agentes apresentam para chegar a um determinado alvo, e suporta variação no número de agentes ativos em campo através do mecanismo de *slots* explicado na secção 7.5.

Durante uma jogada que resulte de uma *set piece* (i.e., todas excetuando a *playNormal*), a situação de jogo encontra-se minimamente definida. Como tal, no momento em que houver uma transição de jogada é executado um algoritmo que determina qual é o melhor *slot* (e, consequentemente, melhor papel) a atribuir a um determinado robô (figura B.8). Cada *slot* tem um papel associado, que depende da jogada ativa. Realizou-se uma avaliação dos papéis existentes nos *slots* de cada jogada, e definiu-se um alvo para o cálculo da função custo, podendo esta ser a distância à bola ou à baliza da própria equipa. Desta forma, a lista é percorrida, é escolhido o agente ativo que apresenta o menor custo (funções *whoIsClosestToBall* e *whoIsClosestToOurGoal*), é atribuído o respetivo *slot* ao agente e remove-se o mesmo da lista (para não lhe voltar a ser atribuído um segundo *slot*). Se a qualquer momento a lista ficar vazia, significa que todos os agentes ativos em campo já têm um *slot* atribuído, parando-se assim a execução do algoritmo. No caso de haver uma variação do número de agentes ativos durante a jogada, este algoritmo é novamente executado e volta a ser feita uma reavaliação de quais os agentes que apresentam um menor custo para ocupar um determinado *slot* e, consequentemente, menor custo para ocupar o papel correspondente ao *slot*.

Por outro lado, a situação de jogo na jogada *playNormal* pode não estar tão bem definida, e esta pode estar ativa mais tempo do que as jogadas que resultam de situações de *set pieces*. Deste modo, para a jogada *playNormal* a avaliação do melhor agente para ocupar um determinado *slot* é realizada em cada ciclo de controlo do programa principal. Este algoritmo (figura B.9) é muito semelhante ao apresentado anteriormente para as restantes jogadas. Contudo, como o primeiro era executado apenas durante as mudanças de jogada ou variações de agentes ativos em campo, garantia que não haviam transições muito frequentes de papéis durante a jogada. Isto podia ser problemático neste segundo algoritmo, visto que é executado em todos os ciclos de controlo, podendo resultar em múltiplas transições de estado no caso de dois agentes estarem a distâncias muito próximas do alvo. Por esse motivo, adicionou-se um incentivo/desconto, medido em distância, que reduz o custo no caso do robô já ter aquele *slot* atribuído, evitando assim, através desta espécie de histerese, transições múltiplas entre *slots*. O incentivo é realizado com base no *slot* do agente e não com base no seu papel ativo de modo a não interferir com a máquina de estados que corre no interior de cada *slot* para uma respetiva jogada (figura 7.7). Nos restantes aspetos, o algoritmo é semelhante ao das restantes jogadas.

Em qualquer dos casos, devido às regras da competição, o primeiro *slot*, que contém sempre papéis correspondentes ao guarda-redes, é atribuído unicamente ao guarda-redes e apenas é atribuído a jogadores com o número 1.

No final de execução destes algoritmos, é realizada a atribuição dos papéis correspondentes a cada *slot* aos agentes.

7.7 Comunicações

As comunicações são um dos pontos chave para atingir uma boa coordenação de uma equipa. Sem comunicações, os agentes não conseguiriam partilhar informação de modo a melhorar o seu *worldstate* e, qualquer tipo de cooperação estava completamente comprometida, deixando apenas a possibilidade de ter agentes a operar individualmente.

No futebol robótico, as comunicações são igualmente um requisito importante que todas as equipas devem incorporar nos seus agentes. Estando perante um ambiente tão alargado e dinâmico como é um campo de futebol durante um jogo, é necessário que os agentes sejam capazes de partilhar informação com os restantes companheiros de equipa, de modo a conseguirem construir uma boa representação do mundo que os rodeia, com a melhor estimativa de bola e obstáculos possível. Da mesma forma, para que exista cooperação, é necessário que os agentes tenham conhecimento da localização e papéis dos restantes companheiros de equipa, seja para decidir a distribuição de papéis pelos agentes, para realizar passes, entre outros.

Particularizando para o caso da *Middle Size League*, as comunicações entre as *Base Stations* e/ou entre os agentes é realizada via *wireless*, respeitando as restrições já apresentadas na secção 2.3.1. Como o número de agentes em campo pode variar, as soluções propostas pelas equipas devem ser suficientemente robustas, permitindo perdas e reconexões de agentes a qualquer instante. Para além disso, como esta é uma aplicação em tempo real, os requisitos temporais a cumprir são bastante restritos, não sendo aconselhável que aconteçam bloqueios na receção de pacotes incompletos quando outros mais recentes já se encontram disponíveis.

No caso da equipa 5DPO, como a cooperação está maioritariamente ao encargo de um supervisor (o treinador), é este o responsável por garantir a partilha de informação entre os agentes da equipa. Como se pode ver na figura 7.8, todos os agentes informam periodicamente o treinador acerca da sua posição atual e destino, velocidades lineares e angulares, estado do sensor de bola, tensão das baterias, estado do robô (ativo ou desativo), tarefa em execução, assim como posição e confiança da sua estimativa de bola. Por sua vez, o treinador constrói a sua representação do mundo e envia a todos os agentes da sua estimativa de bola, jogada ativa da equipa, o papel a desempenhar pelo agente, e a posição e papéis de todos os agentes. Estes dois últimos campos são úteis para que cada agente consiga construir uma representação do mundo que lhe permita identificar e localizar companheiros de equipa em situações de cooperação (passes, formações, entre outros).

Antes do início desta dissertação, o *software* de decisão comunicava com o antigo treinador através de comunicações UDP. A informação a ser enviada era armazenada numa estrutura de dados, que era posteriormente carregada para um *buffer* e enviada para o recetor através de pacotes UDP. Do lado do recetor, o *buffer* recebido era copiado diretamente para uma estrutura de dados exatamente igual à do emissor. Como as duas aplicações estavam desenvolvidas na mesma linguagem e corriam na mesma plataforma (Linux 32 bits), o sistema funcionava na perfeição, e até apresentava bons desempenhos a nível de otimização. Contudo, se a estrutura de dados não fosse exatamente igual, ou se as duas aplicações estivessem a correr em sistemas operativos com

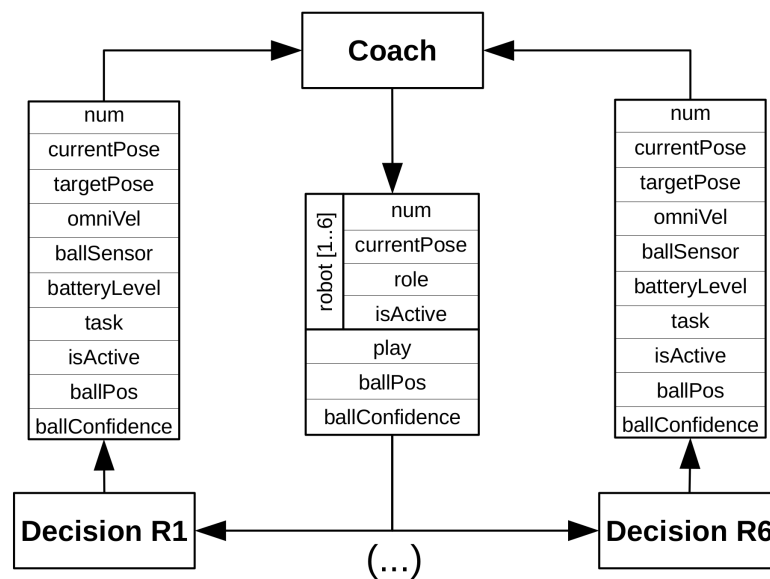


Figura 7.8: Diagrama de Comunicações entre o treinador e o *software* de decisão

arquiteturas diferentes, havia o risco de que partes da mensagem não fossem armazenadas com sucesso. Da mesma forma, sempre que se realizasse uma modificação na mensagem a enviar no emissor, teria de se fazer exatamente a mesma modificação na estrutura de dados do recetor. Por estes motivos, procurou-se uma solução que fosse não só fiável e robusta como a solução anteriormente aplicada, mas também que oferecesse uma maior flexibilidade e independência entre o treinador e o *software* de decisão.

7.7.1 ZeroMQ

ZeroMQ é uma *framework* que fornece *sockets* capazes de transportar mensagens sobre diversas camadas de transporte (no mesmo processo, entre processos, TCP e *multicast*). Esta *framework* permite ter ligações N para N com padrões de comunicação como publicador-subscritor e pedido-resposta. É uma biblioteca leve, que consome poucos recursos e de fácil utilização. Algumas vantagens do ZeroMQ são o facto de tratar as entradas e saídas assincronamente em *threads*, não necessitando de semáforos ou estados de espera; ter uma gestão dinâmica dos componentes reconectando-se automaticamente, permitindo ter serviços a entrar e sair da rede a qualquer instante, sem necessidade de conexão por uma determinada ordem; armazenar automaticamente as mensagens numa pilha quando necessário; conseguir lidar com múltiplas ligações em simultâneo; não impor nenhum formato para as mensagens ou fazer uma gestão de erros automática. Esta *framework* foi desenvolvida em 2007 pela empresa iMatix [70] e é um projeto *open source* distribuído sob licença LGPLv3. A biblioteca foi desenvolvida em C e C++, contando já com cerca de 50 *bindings*, incluindo Python, Java e FreePascal [71, 72].

Em 2011, no âmbito do projeto *Controls Middleware*, o CERN publicou um artigo comparativo de sete tecnologias *middleware* [73]. O objetivo principal era unificar a camada *middleware*

usada nos aceleradores, que é responsável pela comunicação com os servidores que operam os sensores e atuadores dos mesmos. Foram realizados quatro testes: chamadas *request-reply* de 4000 mensagens por segundo com uma *payload* de 4 bytes; chamadas *request-reply* de 5 mensagens por segundo com uma *payload* de 10MB; publicação de 400 mensagens de 8 bytes a 10 clientes em menos de 100ms e publicação de 30 mensagens de 8 bytes a 10 clientes em menos de 20ms. Destes testes, destaca-se o terceiro, onde o ZeroMQ apresentou um tempo de publicação das mensagens aproximadamente constante em função do número de clientes (figura 7.9). Para comparar as diferentes tecnologias tomaram-se como critérios os padrões de comunicação disponíveis, QoS (qualidade de serviço), recursos utilizados, performance, facilidade de uso e suporte da comunidade. No final, a *framework* ZeroMQ obteve a classificação 6/6, sendo selecionada como uma das três *frameworks* a passar à fase de testes e acabando por ser a solução adotada no ano seguinte [74].

Para além do CERN, o ZeroMQ é também utilizado por empresas como a Cisco, NASA, Spotify, Samsung Electronics e Microsoft® [75].

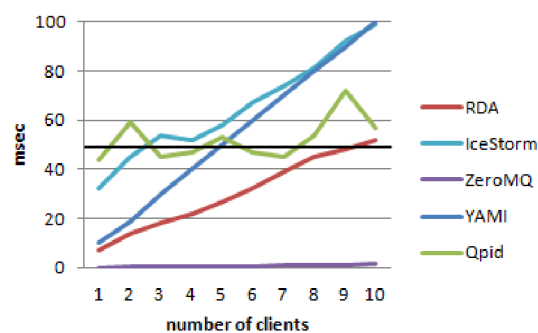


Figura 7.9: Teste 3, publicação de 400 mensagens de 8 bytes a 10 clientes [73]

7.7.2 Implementação

Devido à elevada robustez, simplicidade e principalmente pelo facto de suportar múltiplas conexões e desconexões sem nenhuma necessidade de respeitar uma determinada ordem, decidiu-se utilizar a *framework* ZeroMQ para realizar as comunicações entre o treinador e os agentes da equipa.

Para isso, foi utilizado um *binding* de ZeroMQ para o compilador Free Pascal ², assim como dois componentes para Lazarus que visavam facilitar a interface de utilização e criar uma maior independência entre o emissor e o recetor. O primeiro é responsável por fornecer uma interface que implementa os procedimentos base para o estabelecimento de uma (ou várias) conexões por ZeroMQ. O segundo fornece uma camada extra de abstração, permitindo enviar vários tipos de dados em pacotes, com um identificador associado ao mesmo. Desta forma, a escrita e leitura de mensagens por parte do emissor e recetor fica muito mais transparente, permitindo que o emissor

²<https://github.com/DJMaster/zeromq-fpc>

publique mais mensagens do que o recetor está à espera de receber, bem como uma ordem de leitura que não seja igual à de escrita dos dados.

Perante o fluxo de informação presente na figura 7.8, decidiu-se usar o padrão de comunicação publicador-subscritor. Desta forma, cada agente publica a sua mensagem num tópico em *broadcast*, sendo estes tópicos subscritos pelo treinador. Da mesma forma, o treinador publica a sua mensagem num tópico em *broadcast*, sendo esta subscrita por cada agente.

A ordem do lançamento das aplicações é irrelevante, tendo sido realizados com sucesso vários testes a simular a conexões e desconexões consecutivas de agentes.

7.8 Conclusões

Neste capítulo foi apresentado um supervisor que, embora tenha sido aplicado no âmbito da coordenação de uma equipa de agentes de futebol robótico, apresenta características semelhantes às desejadas para uma aplicação industrial, busca e salvamento, entre outros. Este supervisor inseriu-se numa arquitetura de controlo com uma estrutura hierárquica, estando este responsável pela determinação da tática e jogada da equipa, assim como pela coordenação e cooperação entre agentes. A coordenação consistia na definição de um conjunto de *N slots* (para o máximo de *N* robôs), atribuição de papéis às *slots* que fossem mais adequados para a situação do cenário de aplicação e atribuição de *slots* aos agentes ativos de modo a minimizar o custo de distância do agente a um ponto alvo. A estrutura apresentada permite ainda suportar com sucesso possíveis variações no número de agentes ativos durante a execução de uma missão.

Desenvolveram-se novas formações e novas formas de cooperação entre os agentes. A reintrodução do conceito de tática de equipa permitiu identificar mais facilmente o cenário durante uma situação de jogo corrido (*playNormal*), oferecendo uma maior flexibilidade aquando da atribuição de *slots* (e consequentemente de papéis) aos agentes.

O desenvolvimento de um novo sistema de comunicações com o padrão de comunicação publicador-subscritor, através da *framework* ZeroMQ e respetivos componentes desenvolvidos pela equipa para Lazarus, permitiu obter uma maior interdependência na troca de mensagens entre o *software* de decisão de cada agente e o supervisor, mantendo a robustez e desempenho da solução anteriormente utilizada.

Para além da coordenação, o supervisor desenvolvido possibilita monitorizar o estado interno do mesmo, assim como dos agentes em campo, permitindo identificar casos de falhas e tomar ações corretivas em agentes que não estejam com o comportamento desejado.

Todo o desenvolvimento realizado foi devidamente documentado, de modo a facilitar a compreensão e posteriores modificações por parte de outros elementos da equipa.

Por fim, pode-se concluir que as modificações realizadas a nível da coordenação da equipa 5DPO contribuíram positivamente no desempenho dos agentes da equipa em campo, e forneceram uma *framework* e estrutura que poderá facilmente ser utilizada e/ou adaptada para várias aplicações de cooperação multi-agente.

Capítulo 8

Fusão de Bola no Treinador

Quando se pretende localizar objetos com sensores presentes em robôs móveis, acrescenta-se o problema da incerteza associada à localização do robô. O problema agrava-se quando se pretende fundir a informação oriunda de múltiplos agentes, visto que cada um tem uma incerteza diferente na sua localização. Para além disso, numa equipa de robôs pode haver variações no número de agentes em serviço, e as medições dos sensores podem ser contraditórias, pelo que cabe ao supervisor o papel de fazer uma validação e pré-filtragem antes de efetuar uma fusão adequada dos dados tendo em conta estas variáveis.

No futebol robótico, devido às elevadas dimensões campo, é impossível que os agentes consigam perceber todos os obstáculos e a bola. Estando os agentes distribuídos ao longo do terreno de jogo, é de todo o interesse que estes partilhem informação entre si, melhorando, assim, a sua representação do mundo que os rodeia.

Na equipa 5DPO, a fusão da informação percebida pelos agentes é realizada pelo treinador. Desta forma, o supervisor tem a responsabilidade de receber as várias estimativas de bola dos agentes ativos da equipa, realizar a sua própria estimativa e informar todos os agentes da mesma.

Neste capítulo serão apresentadas duas abordagens para a fusão da bola pelo treinador, nomeadamente, uma baseada num filtro de Kalman e outra baseada numa heurística. Para o filtro de Kalman, serão estudadas e comparadas diferentes abordagens no que concerne às bolas escolhidas para a fase de correção do filtro.

8.1 Abordagens de Fusão de Bola

No caso de terem uma perceção de bola válida, os agentes da equipa 5DPO enviam-na para o treinador. Devido a erros de observação da bola, localização dos robôs ou até observação de falsos positivos, as bolas observadas pelos agentes diferem umas das outras. Cabe ao treinador a função de fundir as mesmas e fornecer aos agentes uma estimativa de bola passível de ser utilizada como meio de validação com as suas estimativas, ou colmatar a ausência de uma.

Deste modo, identificaram-se dois métodos como possíveis candidatos para realizar a fusão da bola dos diversos agentes. O primeiro consistia a utilização de um filtro de Kalman para realizar a

estimação da posição da bola, e fundir as bolas dos diversos agentes. O segundo método consistia numa heurística, em que se escolhia a bola com maior qualidade.

Para facilitar a fusão sensorial, as bolas recebidas e enviadas pelos agentes e treinador estão referenciadas no referencial global. Deste modo, decidiu-se utilizar o filtro de Kalman apresentado na secção 6.5.

Suponha-se o vetor de estado presente na equação (6.53). Numa primeira fase, o treinador realizava uma estimativa de estado da bola e atualizava a incerteza da mesma, de acordo com as equações (6.54) e (6.32). Numa segunda fase, o treinador atualiza a sua estimativa de estado com as (ou a) observações dos diversos agentes, atualizando o ganho do filtro e a incerteza do mesmo (equações (6.39), (6.38) e (6.40)).

Uma questão relevante que se levanta é que bolas utilizar na fase de correção. Se fosse certo que todas as observações realizadas pelos agentes eram exatas e correspondiam à bola de jogo, não restariam dúvidas que deviam ser utilizadas todas as bolas disponíveis na fase de correção. Contudo, devido aos erros de observação e localização dos robôs, as bolas não convergem e, no pior caso, se pelo menos um agente estiver a observar um falso positivo, a utilização de todas as bolas poderia atualizar a localização da bola para uma posição bastante desviada da posição real. Como os agentes validam a sua bola a partir da bola do treinador, a estimação de uma bola errada poderia prejudicar o desempenho de um agente que até poderia estar a observar a bola certa, mas que foi enganado por uma má observação do seu supervisor.

Como tal, definiram-se três abordagens diferentes para a fase de correção da bola: utilizar as bolas de todos os agentes ativos, utilizar apenas as bolas que estivesse a menos de 1m de distância em relação ao centro de massa de todas as observações (com qualidade positiva) e, finalmente, utilizar apenas a bola que tiver maior qualidade (confiança).

Para além do filtro de Kalman, decidiu-se adicionar à comparação um método mais heurístico que consiste em assumir que a bola do treinador corresponde à bola do agente que tiver maior qualidade.

8.2 Ensaios Realizados

De seguida serão apresentados um conjunto de ensaios obtidos em ambiente real, a partir de observações realizadas nos agentes da equipa 5DPO. Estes encontravam-se ligados ao treinador, de modo a obter-se as observações de todos os agentes aproximadamente nos mesmos instantes de tempo. Estas observações serão utilizadas como dados de teste para diferentes abordagens de fusão, utilizando os mesmos dados em cada abordagem de modo a obter-se uma comparação justa.

Serão apresentadas figuras com a disposição dos agentes em campo, qualidade das bolas e posição das bolas em campo. Nestas últimas, será representada uma estimativa da posição real da bola, de modo a facilitar a interpretação de resultados. Note-se que apenas se utilizou uma estimativa baseada no conhecimento real do posicionamento da bola, devido ao facto de não existir nenhum mecanismo de *ground truth* que permitisse obter uma localização mais precisa da mesma.

Ensaio 1 - Todos os Agentes a Observar Bola Estática (Formação *kickoff*)

Neste primeiro ensaio pretende-se avaliar uma situação base, em que todos os agentes da equipa estão ativos, dispostos na formação *kickoff*, e enviam para o treinador a sua estimativa de bola. Esta encontrava-se parada no centro do campo (posição (0,0)).

A figura 8.1 contém uma representação do cenário de ensaio, capturada na janela principal do treinador. Como se pode verificar nesta figura, os agentes 1 e 2 não conseguem observar a bola, devido ao facto de o agente 3 estar à sua frente. Note-se que nestes casos a bola apresenta uma qualidade negativa. Os agentes 3, 4 e 6 observam a bola com uma qualidade razoável, porém, em localizações distintas.

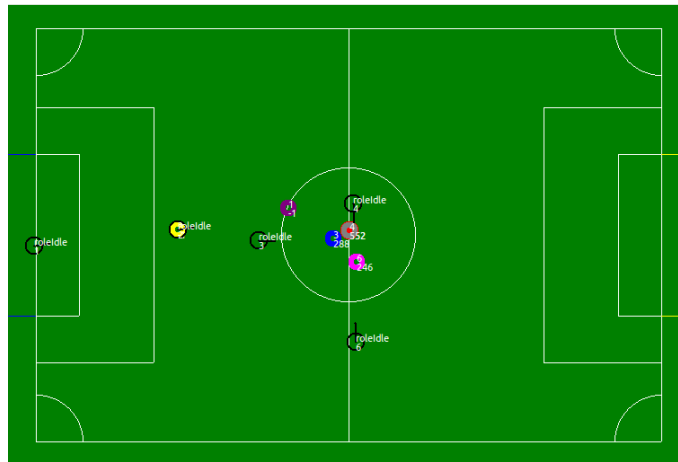


Figura 8.1: Ensaio 1 - cenário de ensaio - todos os agentes ativos

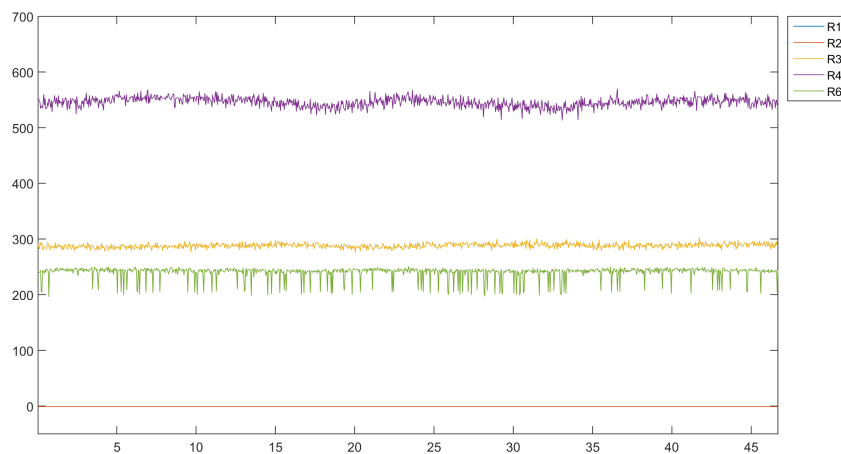


Figura 8.2: Ensaio 1 - qualidade das bolas

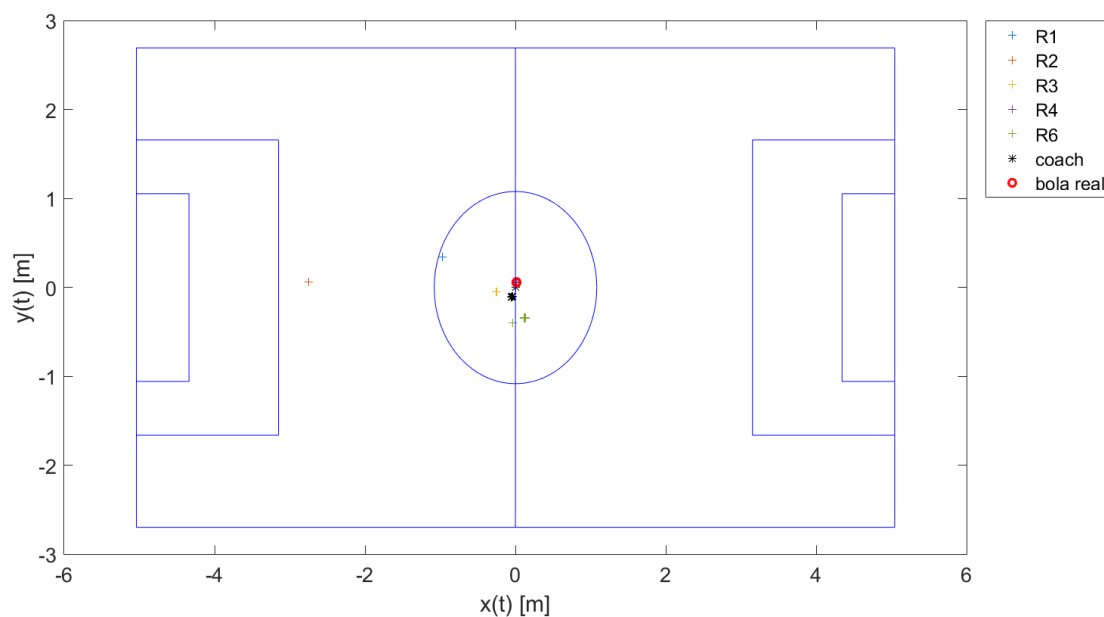


Figura 8.3: Representação no mapa

Figura 8.4: Ensaio 1 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes

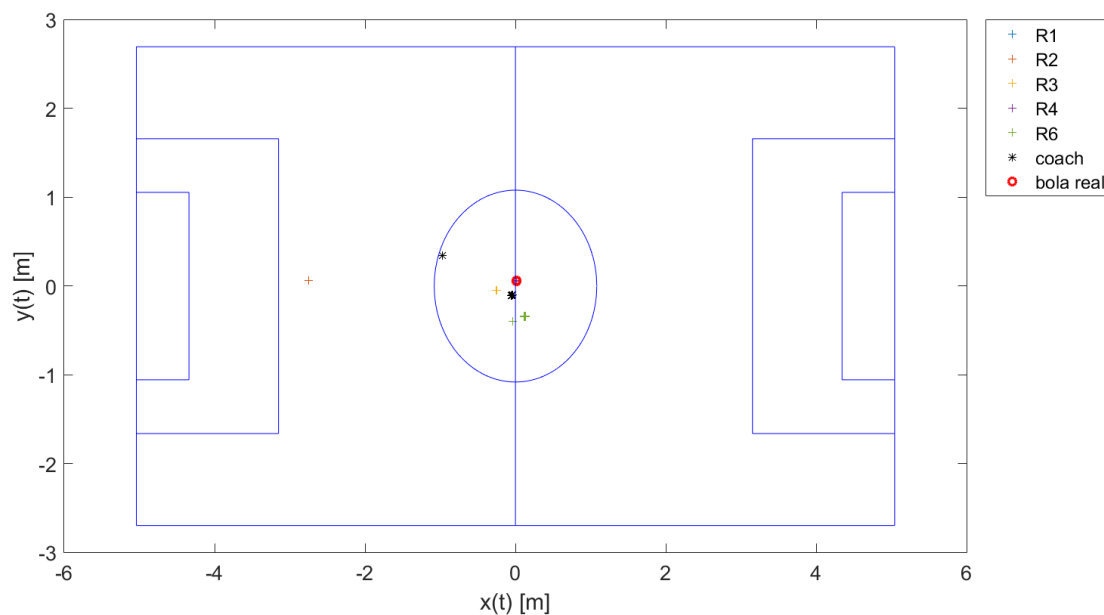


Figura 8.5: Representação no mapa

Figura 8.6: Ensaio 1 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas

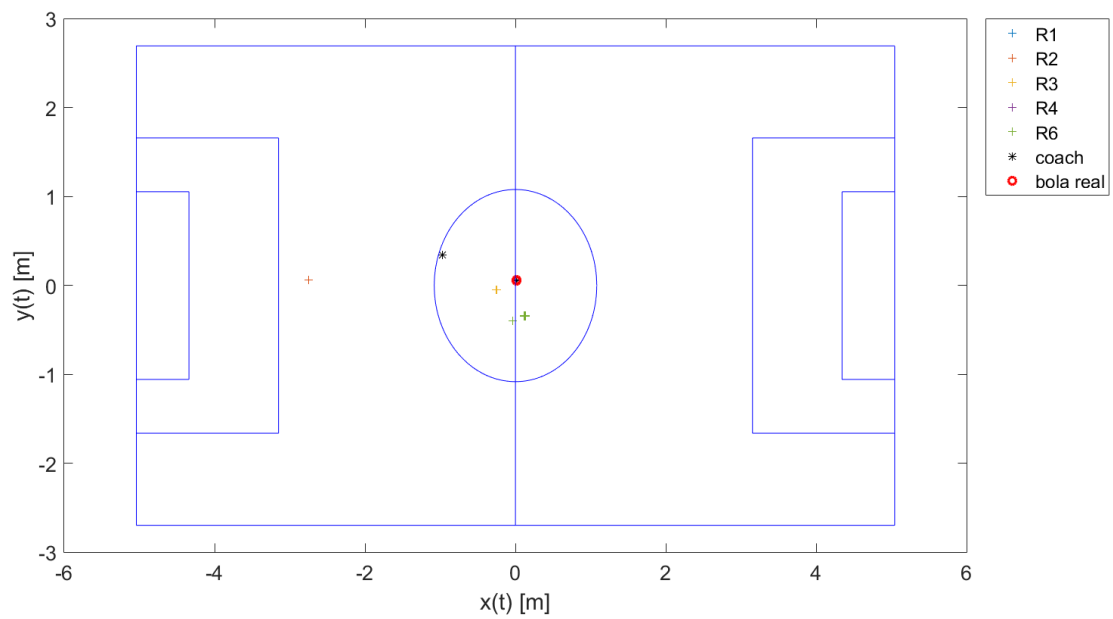


Figura 8.7: Representação no mapa

Figura 8.8: Ensaio 1 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade

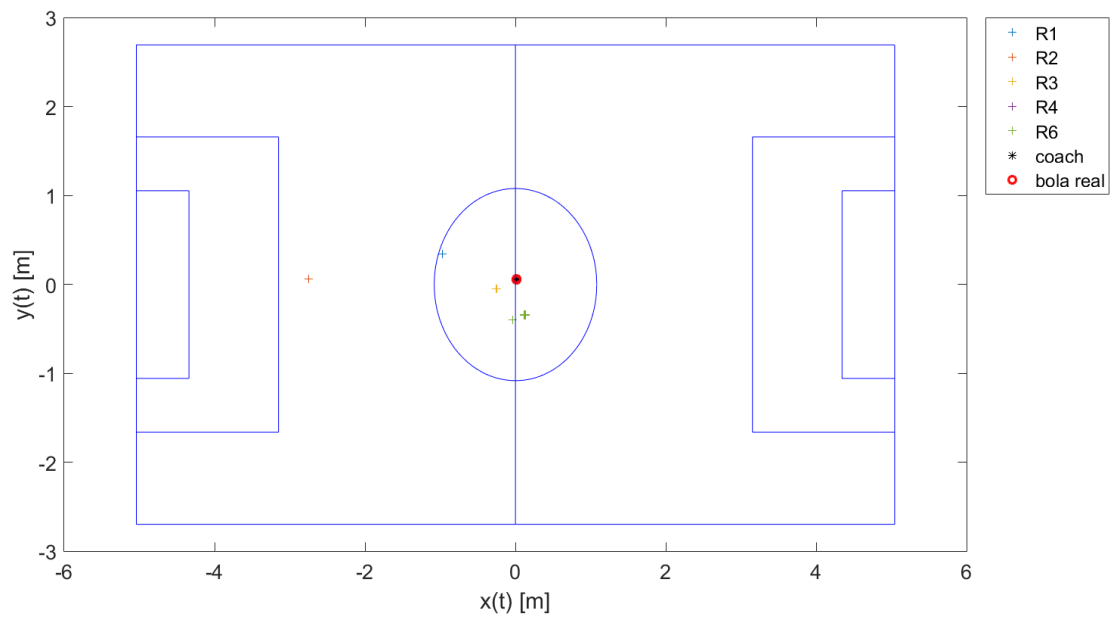


Figura 8.9: Representação no mapa

Figura 8.10: Ensaio 1 - Assunção de bola com maior qualidade

As figuras 8.4, 8.6 e 8.8 contêm os resultados da fusão da bola através de um filtro de Kalman, utilizando na sua fase de correção todas as bolas com qualidade positiva, a bola que se encontrava mais perto do centro de massa de todas as bolas com qualidade positiva, e a bola com maior qualidade. A figura 8.10 contém o resultado da fusão, assumindo que a bola do treinador é a bola que tem maior qualidade.

As abordagens em que se utiliza todas as bolas com qualidade positiva e as bolas a uma distância inferior a 1m do centro de massa das bolas apresentaram o mesmo resultado, visto que, neste caso em particular, utilizavam as mesmas bolas na fase de correção do filtro. Isto deve-se ao facto de apenas as bolas observadas pelos agentes 3, 4 e 6 terem qualidade positiva e, por coincidência, todas as observações estarem contidas num círculo de raio de 1m em torno do centro de massa de todas as observações com qualidade positiva.

Por sua vez, a utilização da bola com maior qualidade na fase de correção e de simplesmente assumir que a bola do treinador é a bola de maior qualidade, também obtiveram resultados semelhantes. Como a bola estava colocada no centro do campo, isto é, aproximadamente no ponto de coordenadas (0,0), é notável que estas duas últimas abordagens obtiveram resultados mais precisos que as duas anteriores. Note-se que a bola que tem maior qualidade é a bola do agente 4, que é aquele que se encontra mais perto da mesma.

Ensaio 2 - Mudança Repentina de Posição da Bola

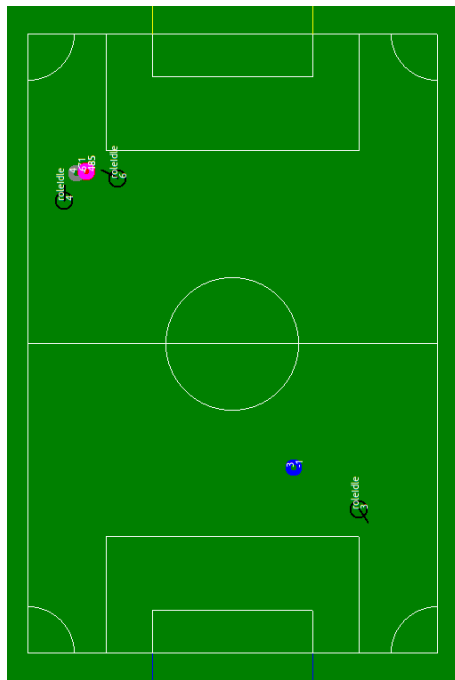
No segundo ensaio, pretende-se testar a situação de uma bola estar a ser observada numa dada localização, repentinamente deixar de ser observada e aparecer posteriormente numa localização diferente. Esta é uma situação que por vezes acontece durante um jogo, por exemplo, quando é marcada uma falta e o árbitro pega na bola e posiciona-a noutro local.

Inicialmente, os agentes 4 e 6 estão a observar uma bola com boa qualidade, enquanto que o agente 3 não obtém nenhuma estimativa de bola. De seguida, a bola é elevada acima da altura das câmaras dos agentes, de modo a que não seja possível a sua observação e, é colocada em frente ao robô 3, de modo que os agentes 4 e 6 não a conseguissem observar. Esta situação pode estar representada na figura 8.11.

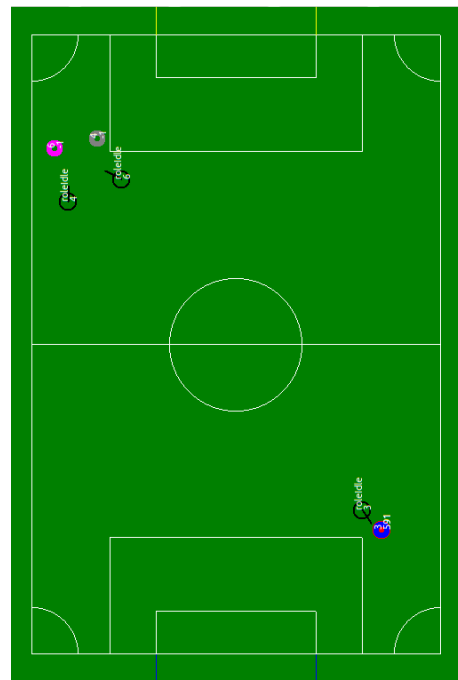
As figuras 8.13, 8.14 e 8.15 contêm os resultados da fusão da bola através de um filtro de Kalman, utilizando na sua fase de correção todas as bolas com qualidade positiva, a bola que se encontrava mais perto do centro de massa de todas as bolas com qualidade positiva, e a bola com maior qualidade. A figura 8.16 contém o resultado da fusão, assumindo que a bola do treinador é a bola que tiver maior qualidade.

Nas abordagens em que se utiliza a bola de todos os agentes e apenas as bolas que se encontra a uma distância inferior a 1m do centro de massa da bola, os resultados são igualmente precisos. Isto deve-se ao facto de, numa primeira fase, a bola estar à mesma distância dos dois agentes que a observam, fazendo com que o uso das duas bolas observadas consiga corrigir o erro de observação dos agentes e, num segunda fase, de apenas estar a ser utilizada uma bola de boa qualidade na fase de atualização (i.e. só o agente 3 é que estava a ver a bola, e esta era de boa qualidade). Mais uma

vez, os resultados são idênticos nesta abordagem porque as bolas utilizadas na fase de correção do filtro de Kalman são coincidentes.



(a) agentes 4 e 6 a observar a bola



(b) agente 3 a observar a bola

Figura 8.11: Ensaio 2 - cenário de ensaio - mudança repentina de posição da bola

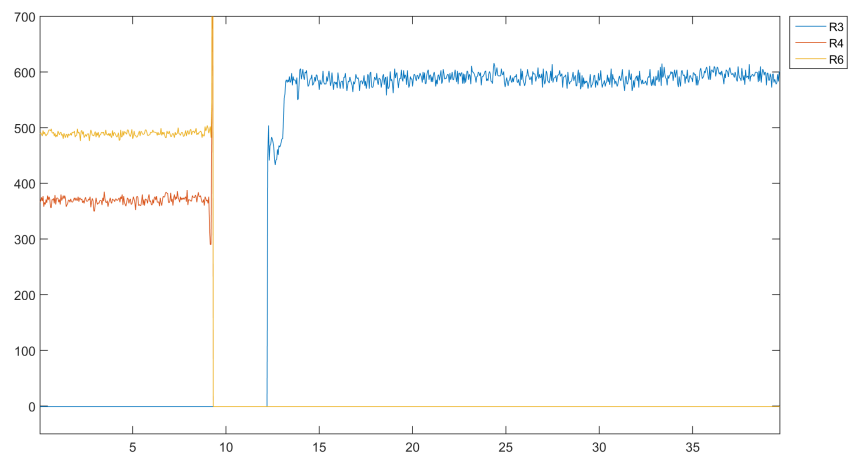
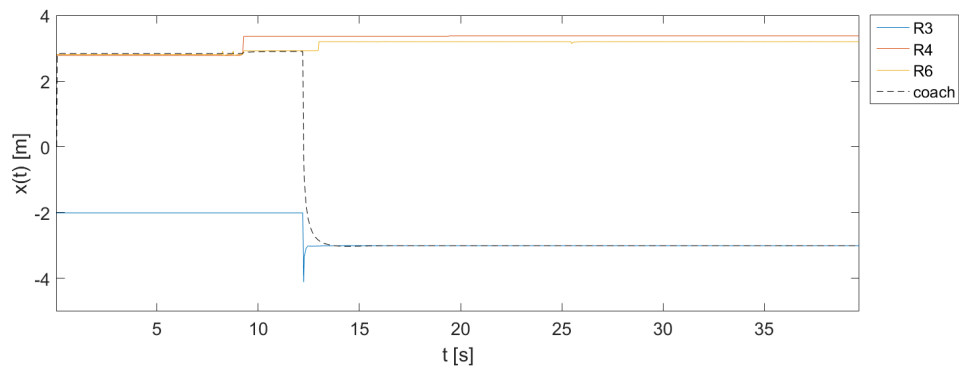
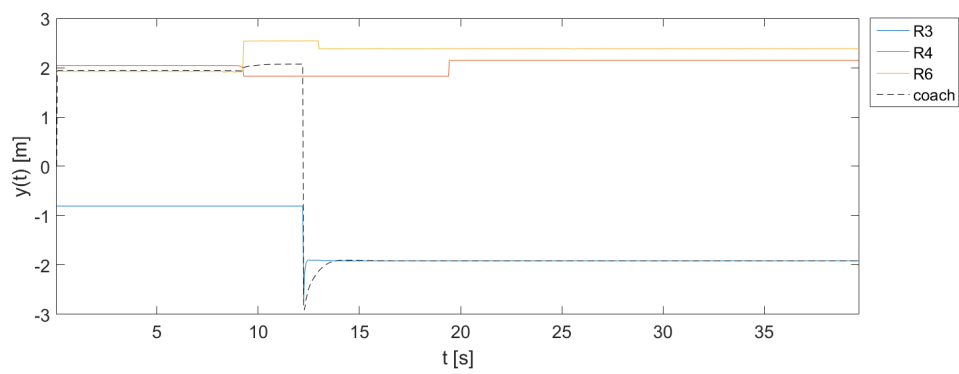
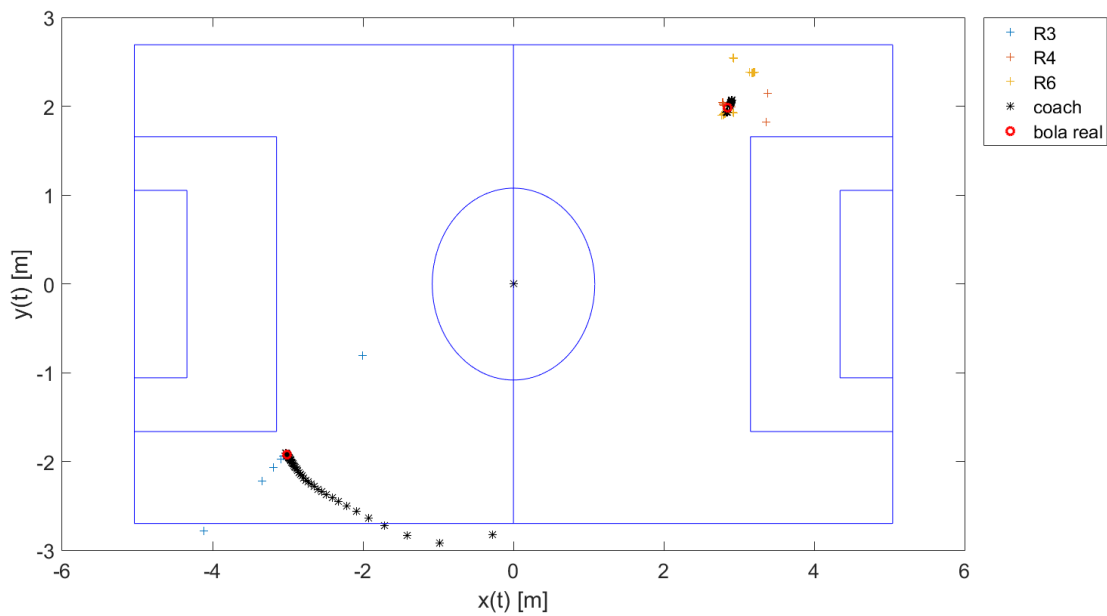
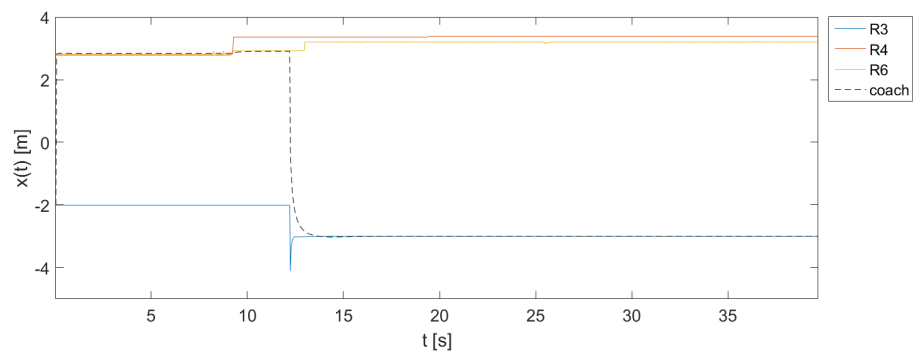
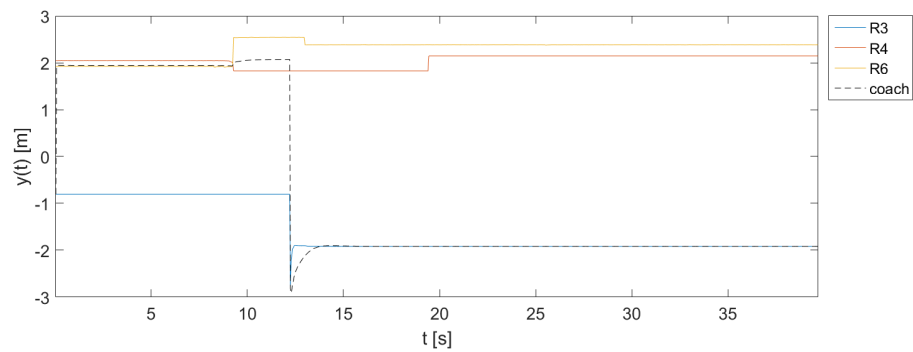
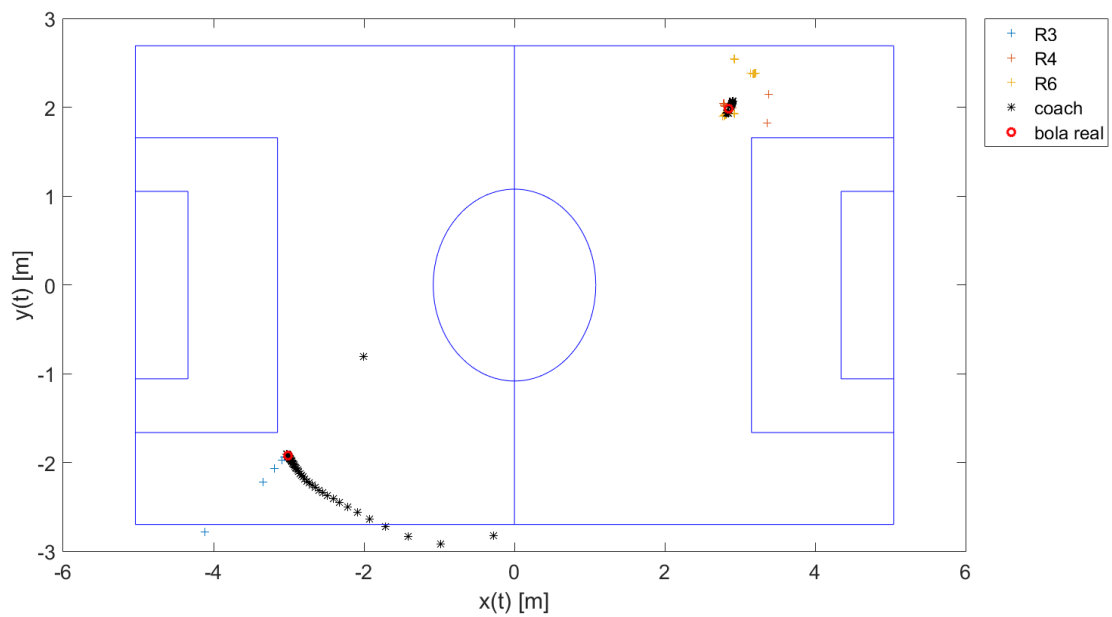


Figura 8.12: Ensaio 2 - qualidade das bolas

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

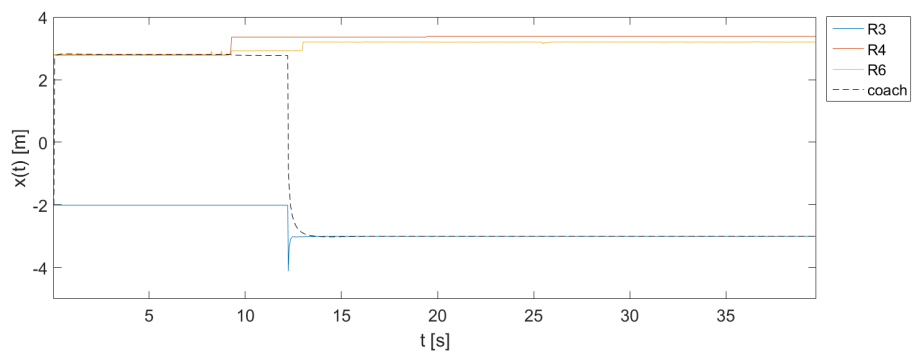
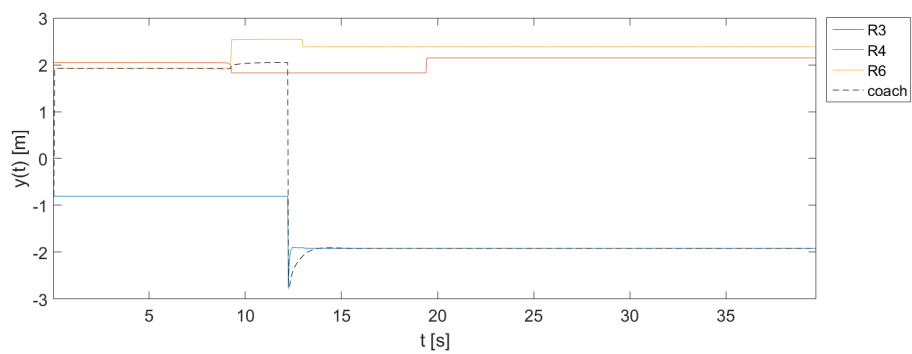
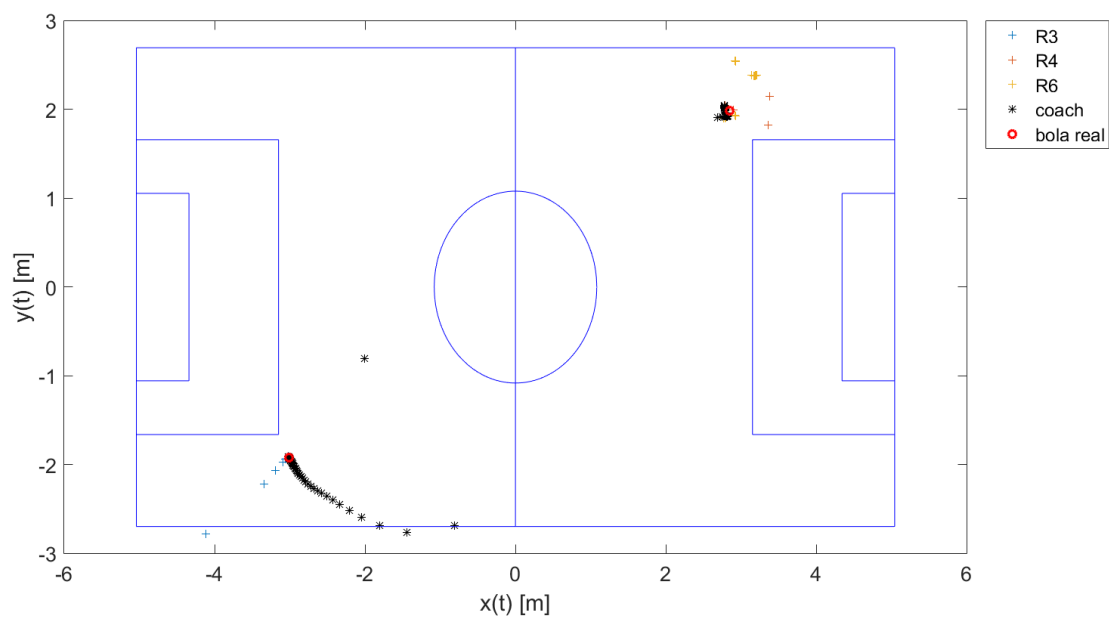
(c) Representação no mapa

Figura 8.13: Ensaio 2 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

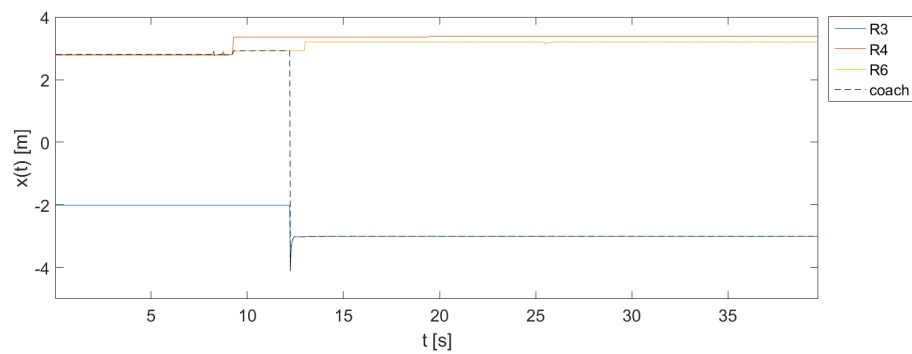
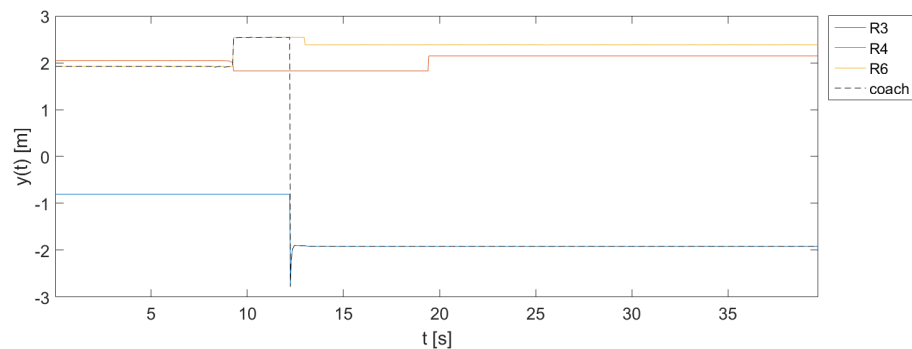
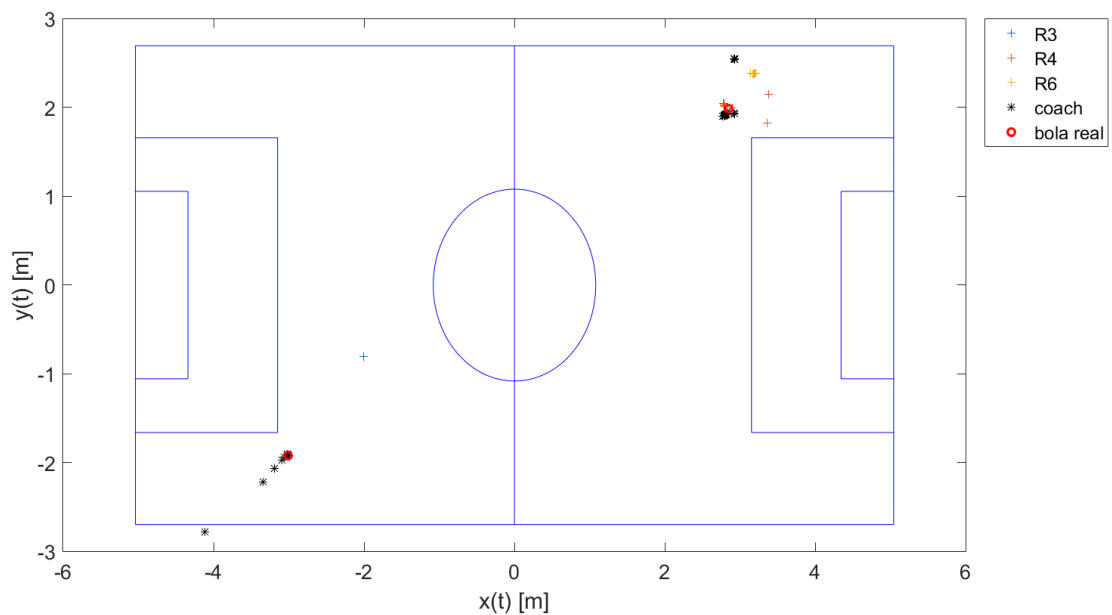
(c) Representação no mapa

Figura 8.14: Ensaio 2 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

(c) Representação no mapa

Figura 8.15: Ensaio 2 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

(c) Representação no mapa

Figura 8.16: Ensaio 2 - Assunção de bola com maior qualidade

Na abordagem em que se utiliza a bola de maior qualidade na fase de correção do filtro de Kalman, os resultados foram muito similares aos das duas abordagens anteriores. Porém, foi visível um pequeno desvio em relação à localização real da bola nos instantes correspondentes à

situação da figura 8.11a, já que, estando a utilizar apenas a bola de maior qualidade, e como a bola se encontrava à mesma distância dos dois agentes, a estimação do treinador estava sempre afetada pelo erro de observação de cada agente.

Na abordagem em que simplesmente se assume que a bola do treinador é a bola com maior qualidade, foram obtidos resultados semelhantes às abordagens anteriores, havendo uma rápida convergência para a localização correta da bola após a mudança de localização da mesma. Todavia, esta abordagem não permite corrigir os erros de observação dos agentes.

Note-se ainda que, durante os instantes em que a bola se encontrou oculta (aproximadamente no intervalo de $t=9s$ a $12s$), as abordagens que utilizam o filtro de Kalman conseguiram manter um estado mais preciso da posição da bola, enquanto que a abordagem que simplesmente utiliza a bola de maior qualidade, embora todas as bolas tivessem qualidade negativa, assumiu a última bola a ter maior qualidade, estando esta numa posição errada. Aqui se nota a vantagem da utilização de um filtro de Kalman nesta aplicação, visto que, enquanto a estimativa de posição de bola dos agentes deva ser o mais precisa possível, dá-se preferência a que a do treinador consiga manter uma estimativa da posição da bola quando esta se encontra oculta. Contudo, esta deve ser realizada com o maior cuidado, de modo a uma transição repentina de posição seja detetada pelo filtro, o que se verificou positivamente neste caso. Outro cuidado que se deve ter, é o tempo que o treinador deverá manter a estimativa de bola, até detetar que esta já não é válida, e dar instruções aos agentes para iniciarem uma procura da bola.

Ensaio 3 - Observação de Falso Positivo

Neste ensaio, pretende-se estudar a situação em que os agentes observam duas bolas com localizações muito distintas. Embora não seja uma situação muito frequente, é passível de acontecer durante um jogo, seja devido ao sol bater diretamente no campo dificultando a classificação de cores, uma peça de vestuário com uma cor e forma semelhante à bola, entre outros.

Desta forma, colocaram-se dois agentes numa extremidade do campo a observar uma bola com boa qualidade, e um terceiro agente noutra extremidade a observar uma segunda bola, também ela de boa qualidade. Alterou-se ligeiramente a calibração de cores do agente 3, de modo a que a bola observada por este tivesse melhor qualidade, assumindo-se que, numa situação de jogo, seria esta a bola "correta", enquanto que a bola observada pelos agentes 4 e 6 seria uma bola falsa. O cenário de ensaio está apresentado na figura 8.17.

As figuras 8.20, 8.22 e 8.24 contêm os resultados da fusão da bola através de um filtro de Kalman, utilizando na sua fase de correção todas as bolas com qualidade positiva, a bola que se encontrava mais perto do centro de massa de todas as bolas com qualidade positiva, e a bola com maior qualidade. A figura 8.26 contém o resultado da fusão, assumindo que a bola do treinador é a bola que tiver maior qualidade.

Como seria de esperar, estando perante três bolas de elevada qualidade em localizações distintas, a utilização de todas as bolas na fase de correção do filtro origina uma estimativa altamente desviada da correta localização de qualquer uma das bolas.

O centro de massa das observações localiza-se aproximadamente no meio campo, perto da posição de coordenadas (0,0). Visto que nenhuma bola se encontra à distância euclidiana de 1m deste ponto, o filtro de Kalman nunca atualiza a sua predição de estado. Consultando a figura 8.22, pode aparentar que esta abordagem leva a uma boa estimativa da posição da bola, porém, isto apenas se deveu à mera coincidência de o filtro ser inicializado com primeira observação do agente ativo com o número mais baixo (que, neste caso, era o agente 3), e pelo facto de a bola ter permanecido parada durante todo o ensaio. Por exemplo, se a bola correta fosse a bola observada pelos agentes 4 e 6, a estimativa desta abordagem estaria completamente errada.

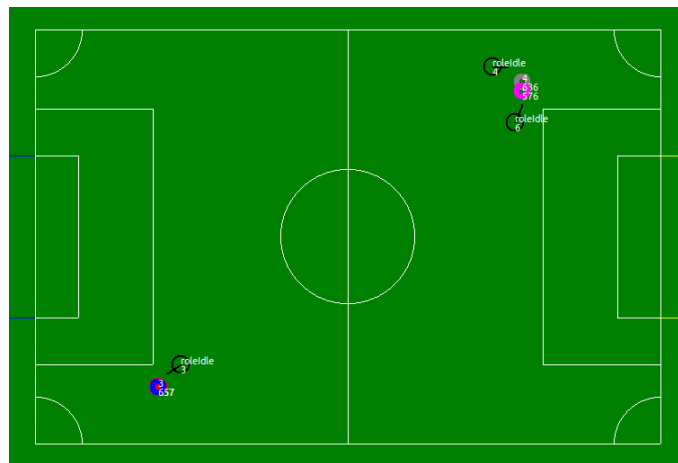


Figura 8.17: Ensaio 3 - cenário de ensaio - falso positivo (duas bolas em campo)

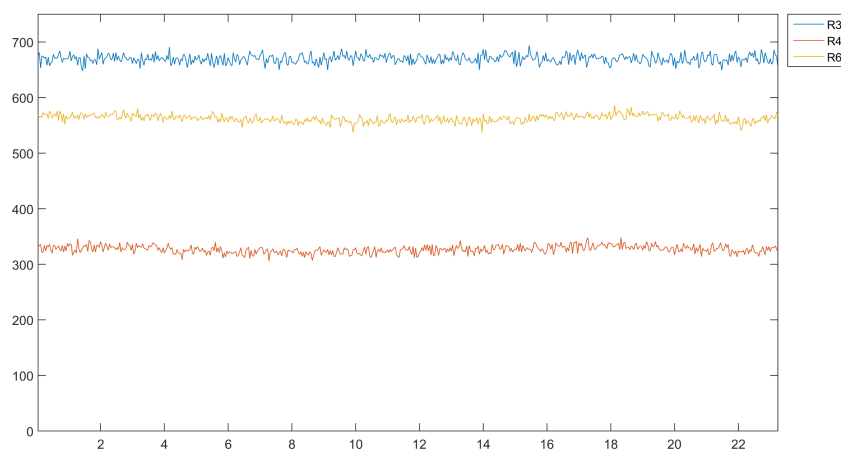


Figura 8.18: Ensaio 3 - qualidade das bolas

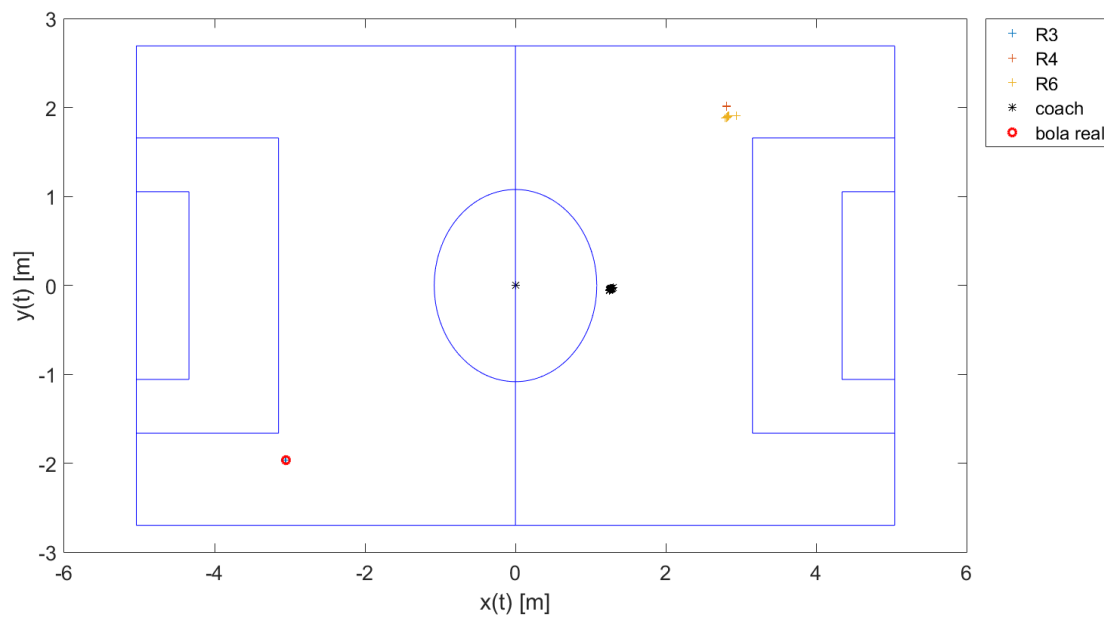


Figura 8.19: Representação no mapa

Figura 8.20: Ensaio 3 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes

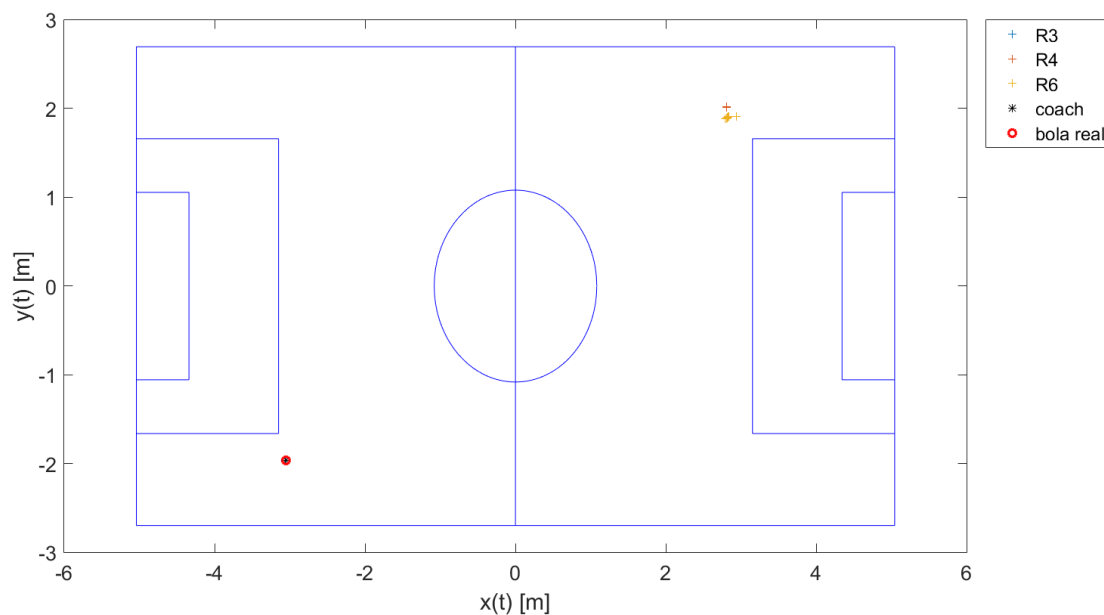


Figura 8.21: Representação no mapa

Figura 8.22: Ensaio 3 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas

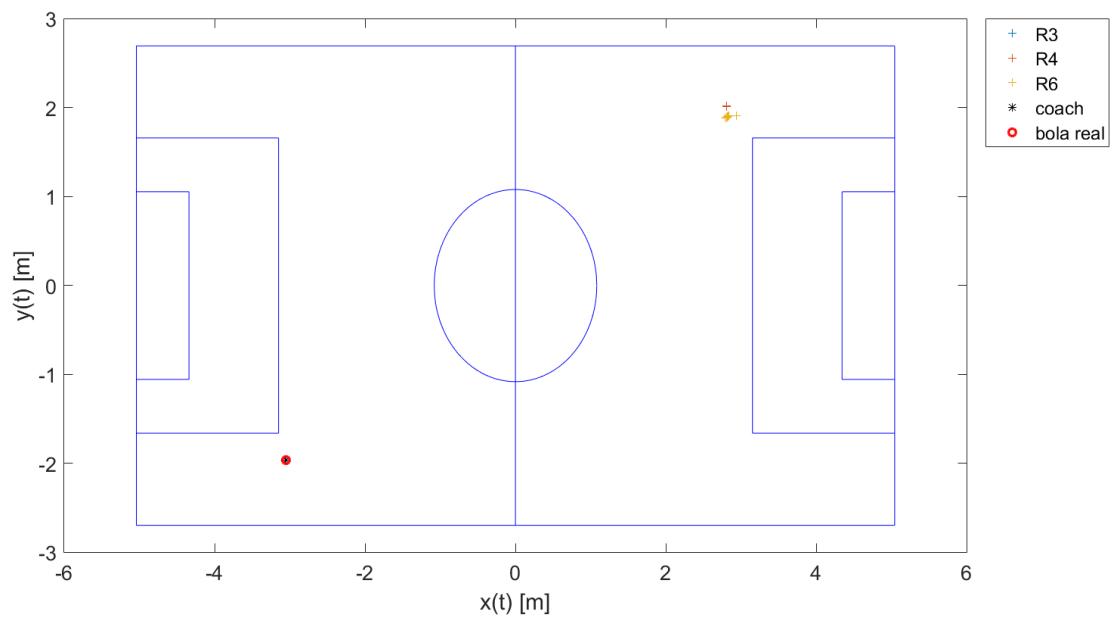


Figura 8.23: Representação no mapa

Figura 8.24: Ensaio 3 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade

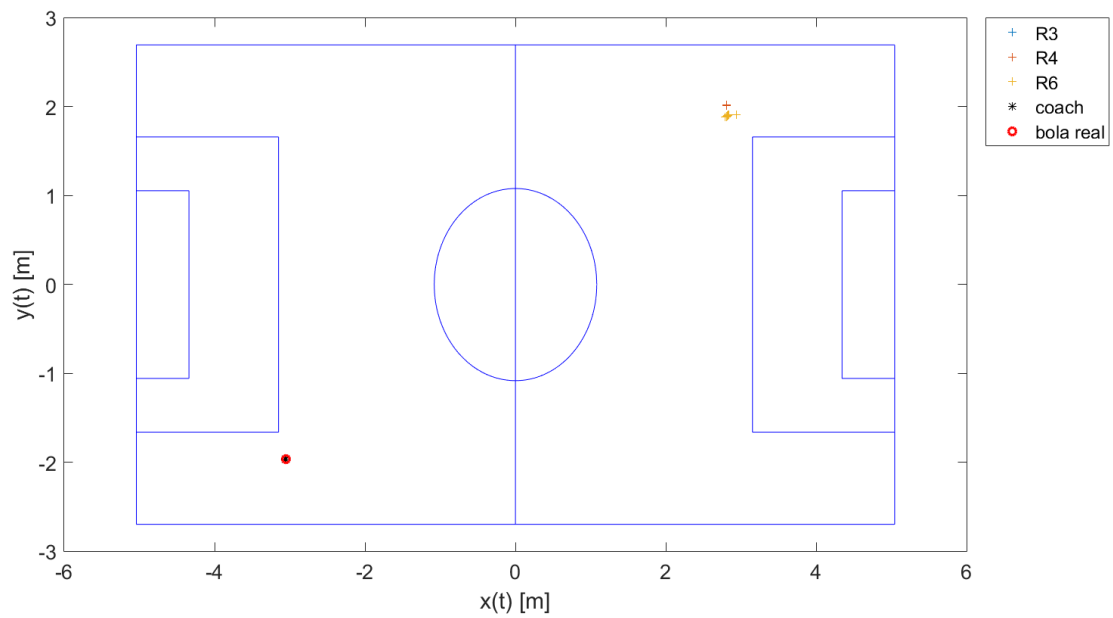


Figura 8.25: Representação no mapa

Figura 8.26: Ensaio 3 - Assunção de bola com maior qualidade

As abordagens que consistem na utilização da bola de maior qualidade na fase de correção do filtro, e da assunção de a bola do treinador é a bola de maior qualidade originam ambos resultados exatos, originando a uma estimativa bastante aproximada da correta localização da bola, apenas afetada pelo erro de observação do agente.

Ensaio 4 - Bola em Movimento

Até ao momento, apenas foram avaliadas situações em que a bola se encontrava parada em relação ao referencial global. Neste ensaio, o cenário é composto por três agentes dispostos em semi-círculo, a observarem uma bola em movimento. A bola partiu da zona da grande área (figura 8.27a), e descreveu um movimento maioritariamente segundo a direção x , parando perto do *grabber* do agente 3 (figura 8.27b).

As figuras 8.29, 8.30 e 8.31 contêm os resultados da fusão da bola através de um filtro de Kalman, utilizando na sua fase de correção todas as bolas com qualidade positiva, a bola que se encontrava mais perto do centro de massa de todas as bolas com qualidade positiva, e a bola com maior qualidade. A figura 8.32 contém o resultado da fusão, assumindo que a bola do treinador é a bola que tiver maior qualidade.

As abordagens que consistiam em utilizar todas as bolas, ou apenas as bolas a uma distância inferior a 1m do centro de massa das observações na fase de correção do filtro de Kalman, originam uma estimativa bastante aproximada da trajetória realmente descrita pela bola. Note-se ainda que a abordagem que utiliza todas as bolas apenas apresentou bons resultados devido ao facto de o centro de massa destas se encontrar perto da localização real da bola.

Analisando a figura 8.28, verifica-se que nem sempre foi o mesmo agente a observar a bola com maior qualidade, nomeadamente no período em que esta se encontrava em movimento. Como tal, a abordagem em que simplesmente se assume que a bola do treinador é a bola com maior qualidade, origina alguns saltos na trajetória e, em grande parte do percurso, originou uma estimativa ligeiramente desviada da verdadeira localização da bola. Porém, no final, acaba por convergir para a posição correta, devido ao facto de o agente 3 ter a bola muito perto de si, originando uma maior qualidade.

Da mesma forma, utilizando apenas a bola de maior qualidade na fase de correção do filtro de Kalman, origina uma trajetória na bola que não corresponde de todo à verdadeiramente descrita pela mesma. Existe a agravante de que esta abordagem origina componentes de velocidade da bola (mais concretamente em v_y), bastante superiores à realmente descrita pela mesma. Isto demonstra que este filtro não deve ser utilizado para estimar velocidades e posições com precisão, visto que esse é o papel do filtro existente em cada agente.

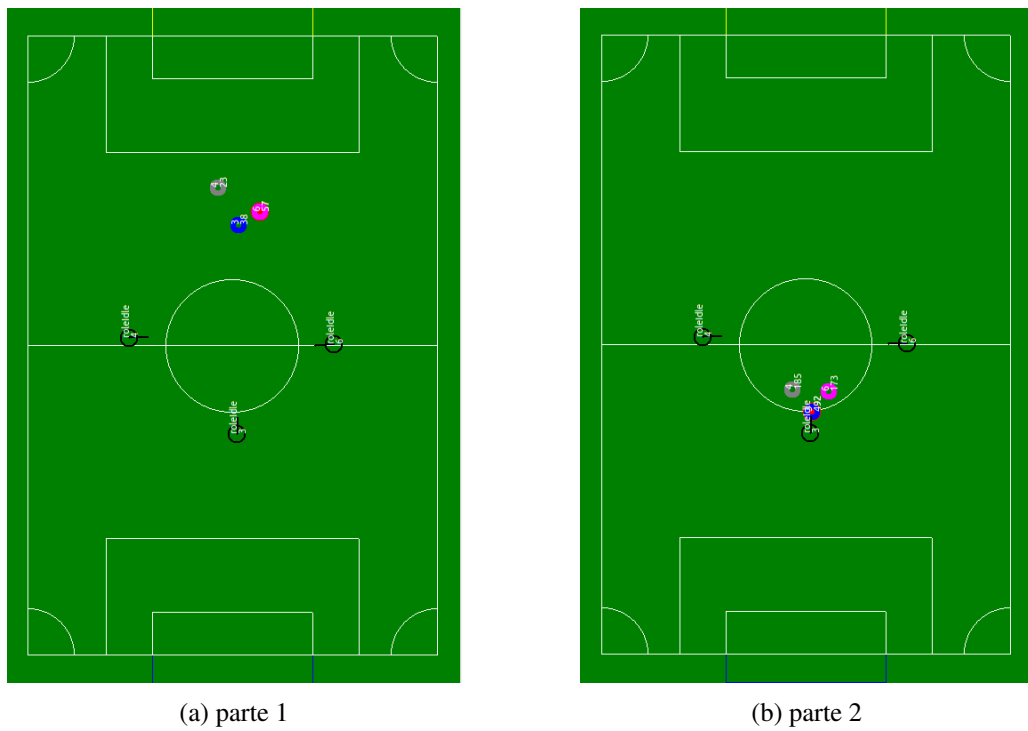


Figura 8.27: Ensaio 4 - cenário de ensaio

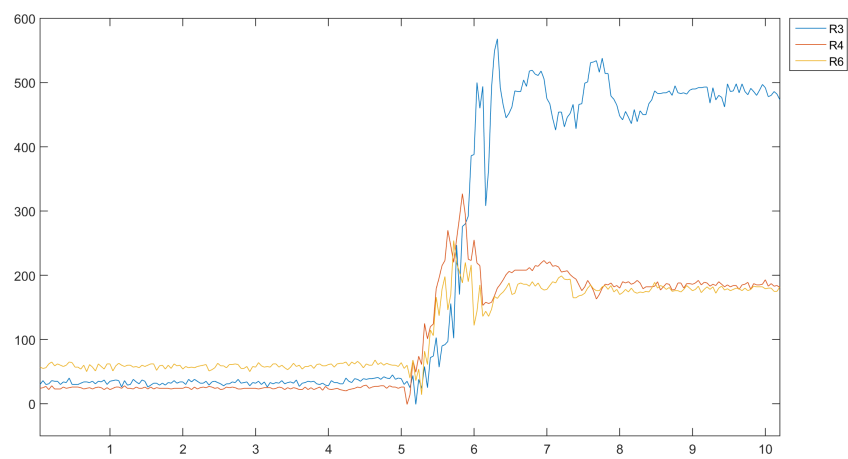
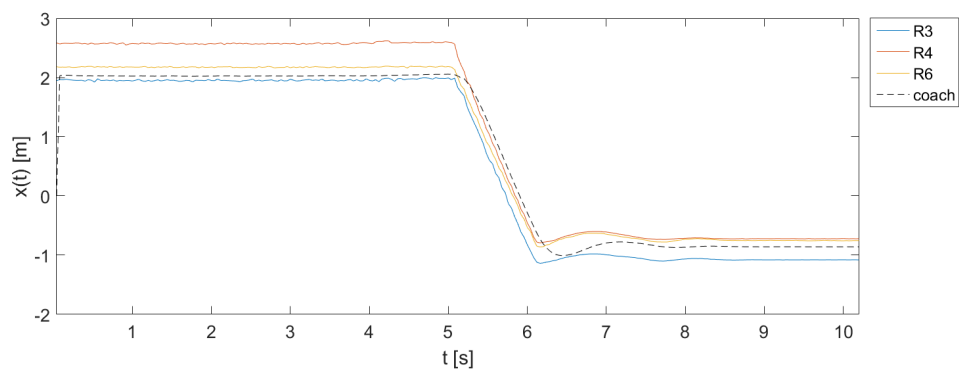
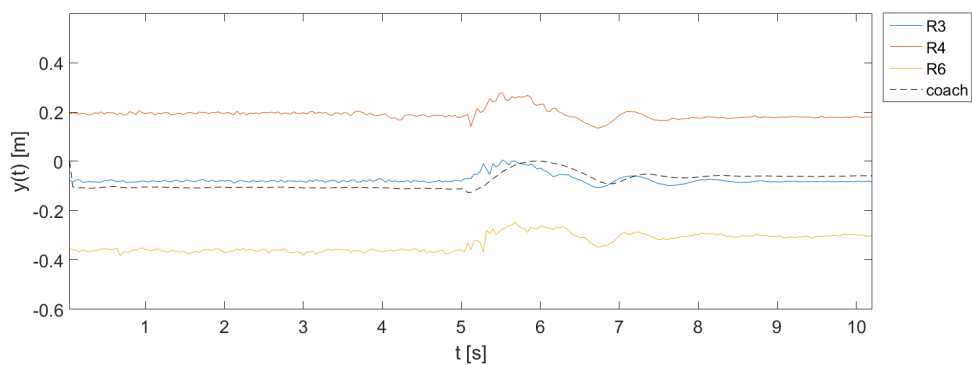
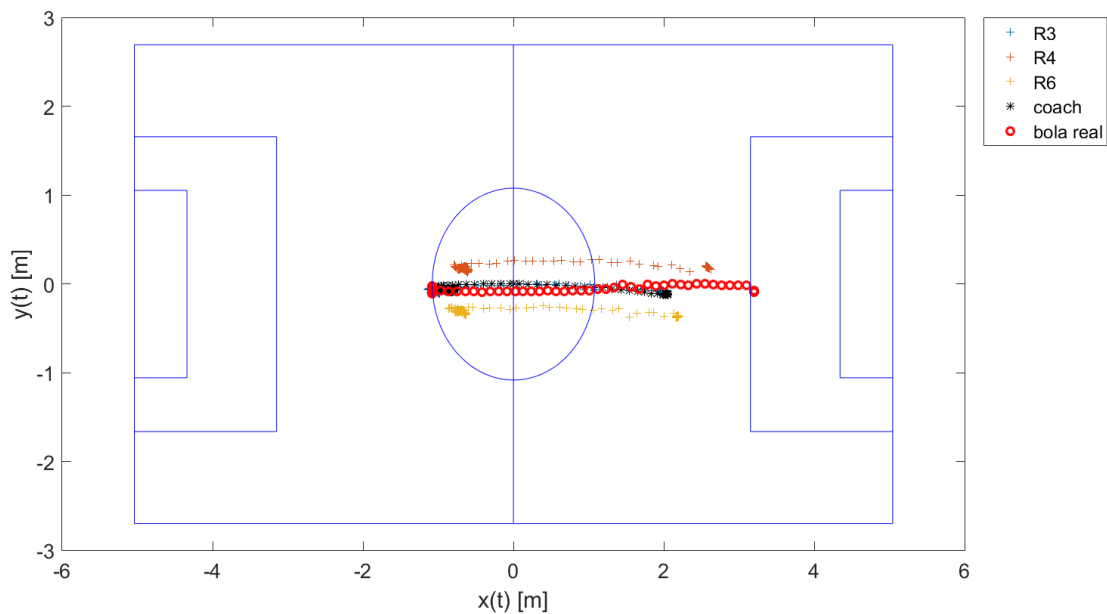
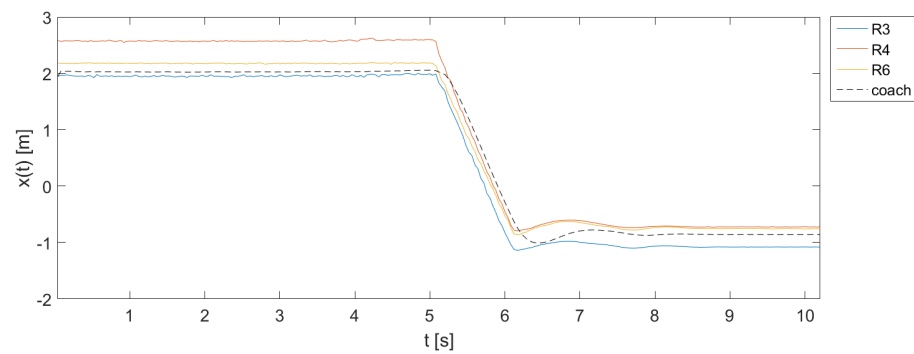
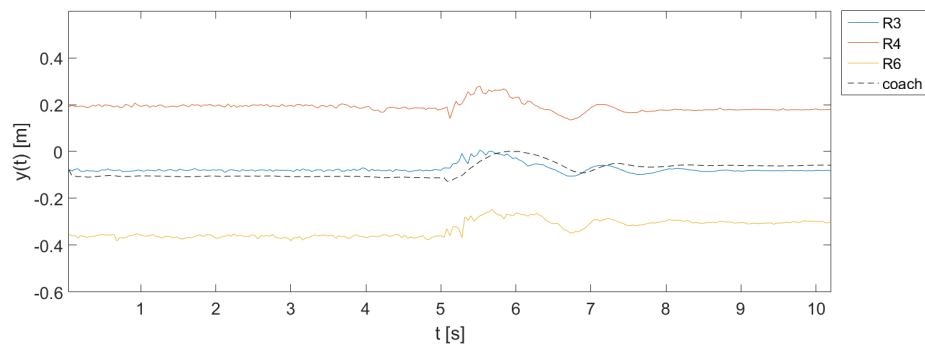
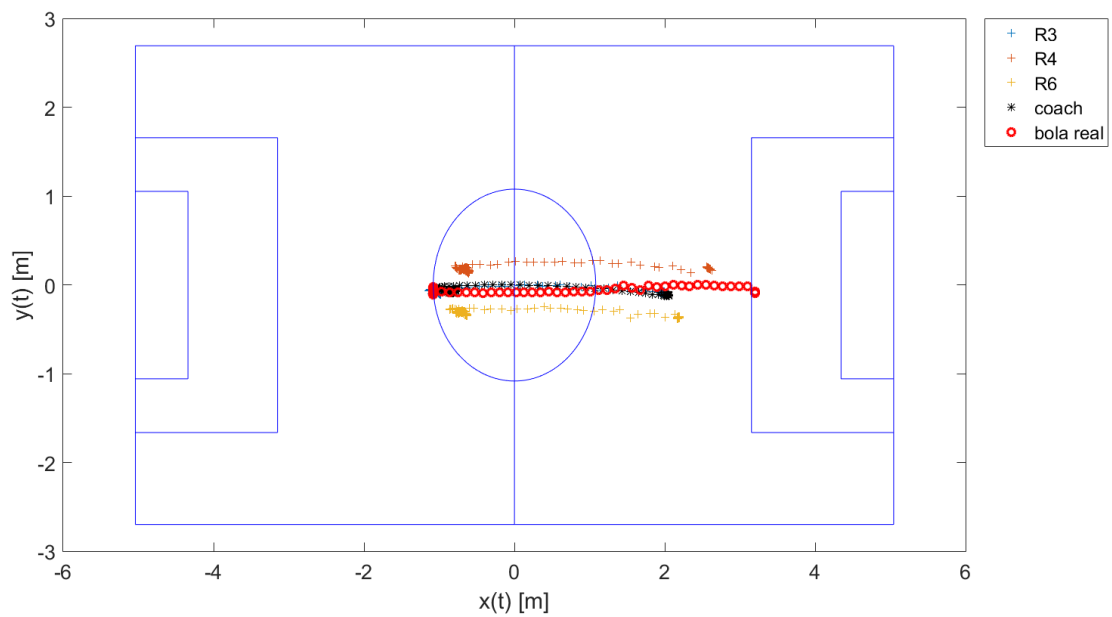


Figura 8.28: Ensaio 4 - qualidade das bolas

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

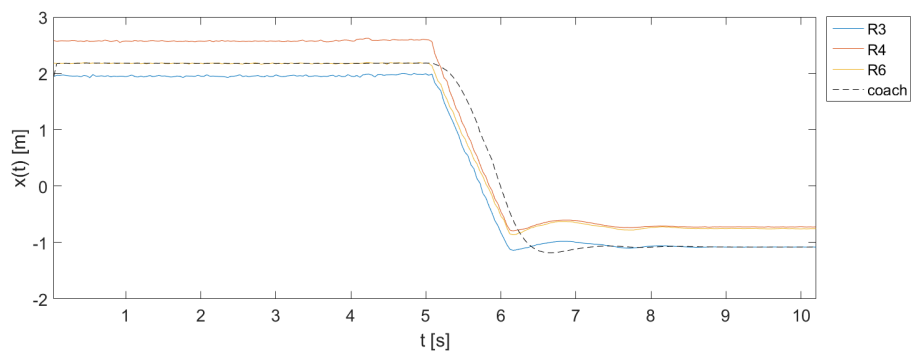
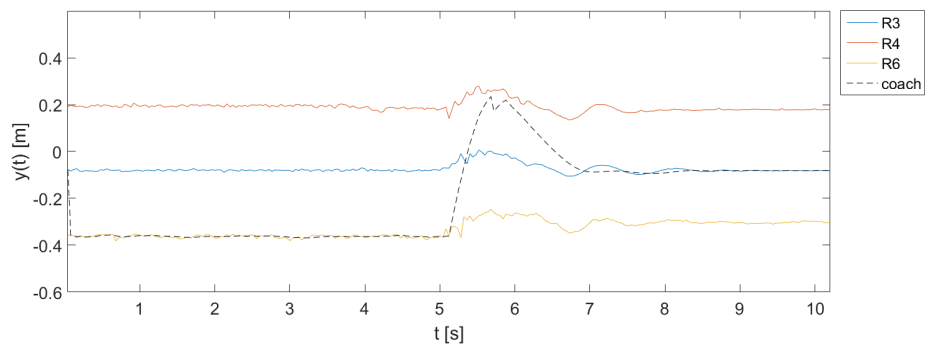
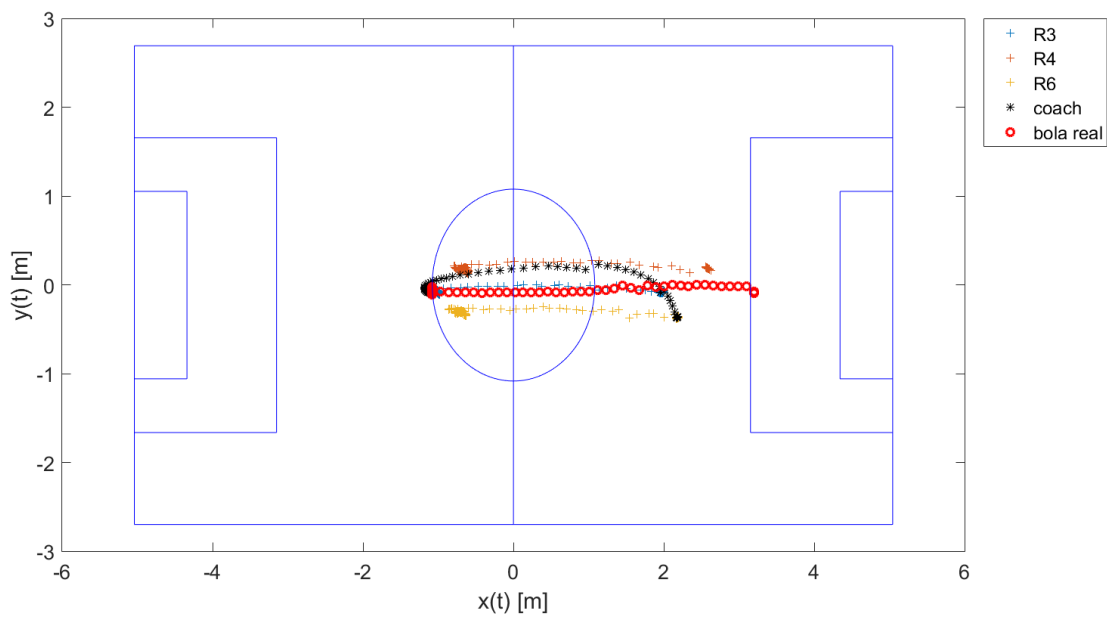
(c) Representação no mapa

Figura 8.29: Ensaio 4 - Filtro de Kalman - fusão utilizando as bolas de todos os agentes

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

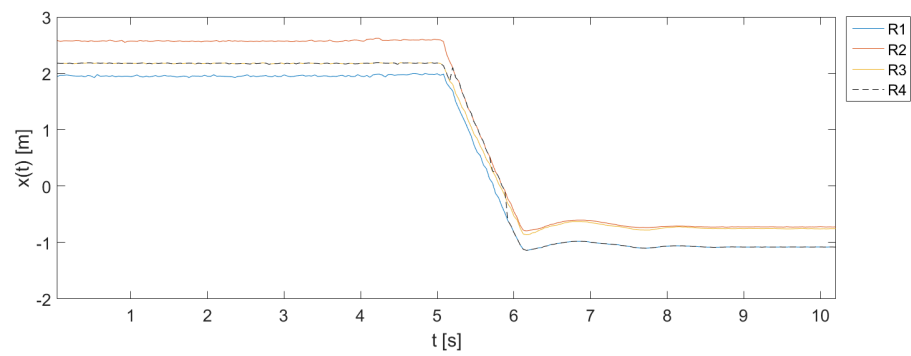
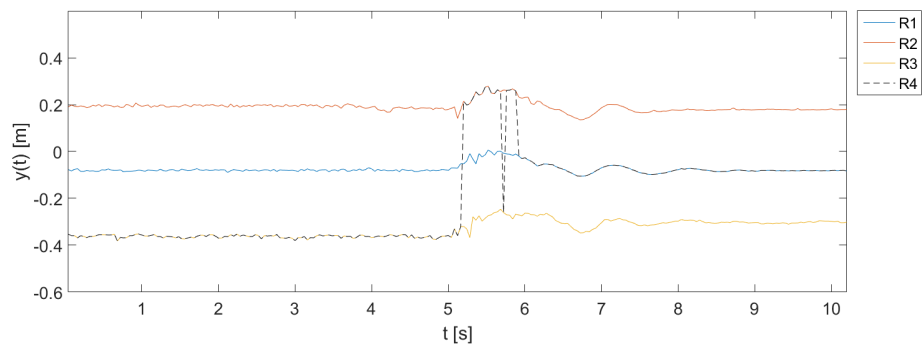
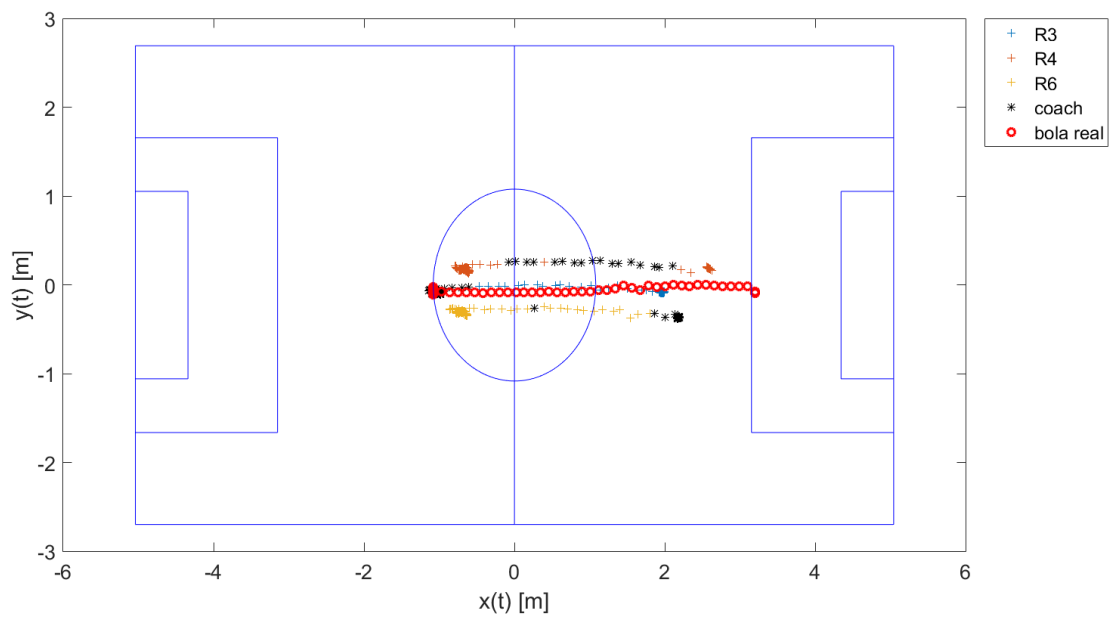
(c) Representação no mapa

Figura 8.30: Ensaio 4 - Filtro de Kalman - fusão utilizando as bolas a menos de 1m de distância do centro de massa de todas as bolas

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

(c) Representação no mapa

Figura 8.31: Ensaio 4 - Filtro de Kalman - fusão utilizando a bola mais com maior qualidade

(a) Posição em x em função do tempo(b) Posição em y em função do tempo

(c) Representação no mapa

Figura 8.32: Ensaio 4 - Assunção de bola com maior qualidade

8.3 Conclusões

Neste capítulo foram apresentadas duas abordagens para a fusão de posição da bola, por parte de treinador. A primeira consistia na utilização de um filtro de Kalman, semelhante ao apresentado na secção 6.5, e foram analisadas três abordagens distintas para a fase de correção do filtro: uma primeira que utilizava todas as bolas observadas que tivessem uma qualidade positiva, uma segunda que apenas utilizava as bolas que estivessem a uma distância inferior a 1m do centro de massa de todas as bolas observadas (com qualidade positiva e depois de um processo de votação), e finalmente uma terceira que apenas utilizava a bola de maior qualidade. Foi também apresentada uma abordagem mais heurística, em que simplesmente se assumia que a bola do treinador era a bola observada que tivesse melhor qualidade.

Foram realizados um conjunto de ensaio, nos quais colocaram à prova diversas situações passíveis de acontecer durante um jogo. Nestes ensaios, foi possível verificar que a utilização de todas as bolas na fase de correção do filtro de Kalman apenas apresenta resultados mais próximos dos exatos quando todas as bolas com qualidade positiva estão muito próximas da bola real, e no caso de nenhum dos agentes estar muito perto da bola. Poderá até levar a estimações completamente erradas, como no caso de serem observadas duas bolas em campo, com localizações completamente distintas. A utilização de apenas as bolas que se encontram a uma distância inferior a 1m do centro de massa de todas as observações aparenta levar a bons resultados, porém contém algum desvio no caso de um dos agentes se encontrar muito perto da bola e, no caso de serem observadas mais do que uma bola de boa qualidade em localizações suficientemente distantes que levem a que nenhuma delas esteja contida no círculo de raio 1m do seu centro de massa, esta abordagem não providencia nenhuma medida para a fase de correção do filtro. A utilização da bola de maior qualidade na fase de correção do filtro de Kalman apresentou maioritariamente bons resultados, principalmente no caso de haver um agente que esteja mais perto da bola. Da mesma forma, a assunção de que a bola do treinador é a bola que tiver maior qualidade também apresentou bons resultados.

Na verdade, devido ao facto de o erro de observação dos agentes ser aproximadamente proporcional à distância que estes se encontram da bola, leva a que o agente que esteja mais perto da bola realize uma medição mais exata. Para além disso, o agente que costuma observar a bola com maior qualidade é também aquele que se encontra mais perto da mesma. Isto leva a que, em geral, a bola que tem maior qualidade corresponda à estimativa de bola mais correta.

Desta forma, pode-se concluir que todas as abordagens apresentaram vantagens e desvantagens. Como durante um jogo podem existir observações de bolas dispersas, a equipa não se pode sujeitar a que um agente que esteja a realizar uma boa observação seja enganado por uma má estimativa do treinador, ou que seja atribuída uma tarefa que envolva interação com a bola a um agente que esteja mais distante que outro. Assim, conclui-se que as melhores abordagens para o contexto em questão são a utilização da bola que apresentam maior qualidade na fase de correção do filtro de Kalman, ou simplesmente a assunção de que essa é a bola do treinador. Note-se que a abordagem que nas abordagens que englobam filtros de Kalman, é possível ter em conta a in-

certeza associada à localização dos agentes e às suas estimativas da bola na estimação da posição da mesma. Da mesma forma, no caso de a bola estar momentaneamente desaparecida, conseguem manter uma estimativa da posição da bola mais correta do que a assunção de que a bola do treinador é a bola de maior qualidade. Contudo, a continuação da estimação de uma bola desaparecida deve ser realizada com o maior cuidado, devendo o filtro ser capaz de detetar o aparecimento da bola numa localização diferente, mas, principalmente, ter um mecanismo de *timeout* que detete que a bola está perdida há muito tempo, de forma a que o treinador coloque os seus agentes a procurar a bola.

É igualmente importante referir que, devido a restrições temporais, apenas foi possível implementar no supervisor a abordagem de assumir que a bola do treinador é a bola com maior qualidade. Note-se que esta foi a solução utilizada na competição Rotótica2017, tendo demonstrado que, no caso de os agentes conseguirem realizar uma boa observação, origina uma boa estimativa de posição de bola por parte do treinador face à real localização da mesma, permitindo atribuir os papéis de uma forma mais adequada. Todavia, sugere-se vivamente a realização de testes das abordagens que utilizam filtro de Kalman no treinador (principalmente a que utiliza a bola de maior qualidade na fase de correção), em ambiente real de jogo, de modo a que sejam testadas mais cenários, e se possa realizar uma análise mais justa entre abordagens.

Capítulo 9

Conclusões

9.1 Trabalho Realizado

Nesta dissertação, começou por se recuperar e melhorar um cenário já existente da equipa MSL de futebol robótico 5DPO para o simulador SimTwo. Adicionaram-se não só características físicas dos robôs reais (mecanismo de *grabber*, sensor de bola, mecanismo que permitisse realizar um *low kick* ou *high kick*, e outras alterações que tornaram a simulação mais leve), como também se refez todo o *script* de controlo, permitindo a interação e comando dos agentes a partir do *software* de decisão da equipa (*Decision*). Este foi um trabalho fundamental, que permitiu obter um ambiente de simulação realista para a implementação e teste de simulações do filtro de bola e da cooperação dos agentes da equipa. Este simulador foi extremamente útil na competição Robótica2017, permitindo testar a implementação dos novos papéis dos agentes, assim como as formações e cooperação da equipa, mesmo quando o campo se encontrava ocupado com jogos ou testes de outras equipas.

De seguida, desenvolveram-se duas abordagens de estimação de posição e velocidade da bola em referenciais móveis, baseadas em filtros de Kalman com os vetores de estado referenciados no referencial local e global. O filtro de Kalman no referencial global revelou-se um bom estimador da posição da bola, porém, devido à sua dinâmica ligeiramente mais lenta, não apresentou resultados tão favoráveis na estimação de velocidade. Foi também apresentada uma ligeira alteração ao filtro, que se revelou capaz de compensar o efeito que uma correção na localização do robô tem na estimativa de velocidade da bola, devido ao facto de a estimação estar a ser realizada no referencial local. Já o filtro de Kalman com o vetor de estado no referencial local, apresentou bons resultados quer na estimação de posição como de velocidade da bola. Para além disso, apresenta a vantagem de ser completamente independente da localização do robô, conseguindo realizar a estimativa independentemente da existência ou não de uma correta localização do robô, permitindo que este consiga realizar ações com/em relação à bola com rigor e precisão. No final, foi ainda realizada uma comparação da estimação de velocidade com base numa regressão linear através de uma regressão linear, consistindo numa simplificação de uma abordagem desenvolvida pela equipa CAMBADA [37, 43]. Concluiu-se que se conseguiam obter resultados semelhantes, ha-

vendo ligeiras vantagens para a abordagem do filtro de Kalman, nomeadamente, estimações mais suaves quando a bola está parada, e uma convergência mais rápida para a velocidade da bola no caso de esta se encontrar em movimento. Contudo, neste último caso, a estimativa da velocidade obtida através da regressão linear apresentou um valor ligeiramente mais exato que a do filtro de Kalman. Para além disso, o filtro de Kalman apresenta ainda a vantagem de, a qualquer momento de tempo, ter uma matriz de covariância da estimativa de estado que representa a confiança que se pode ou não ter sobre cada componente da velocidade da bola.

Desenvolveu-se também um supervisor que se revelou capaz de coordenar uma equipa de robôs móveis. Foram realizados testes em ambiente de simulação e em ambiente real, nos agentes da equipa de futebol robótico médio 5DPO. O supervisor inseriu-se numa arquitetura de controlo com estrutura hierárquica que é responsável por, a partir das instruções do árbitro (recebidas por comandos através duma aplicação denominada *RefereeBox*), determinar a melhor tática, jogada para a equipa e, com base nestas, a melhor formação para a equipa, assim como o melhor papel a desempenhar por cada agente ativo. O supervisor é capaz de suportar variações no número de agentes ativos, introduzindo, para isso, o conceito de *slots*. Em função da jogada, atribuí-se os papéis que sejam mais adequados às situações de jogo às *slots* e, posteriormente, atribuem-se as *slots* (e papel correspondente) aos robôs que estejam em melhores condições de desempenhar o papel (determinado através de um custo em distância). Através da interface gráfica do supervisor, é possível monitorizar o estado do jogo e dos agentes, durante uma partida. O trabalho desenvolvido contribuiu positivamente no desempenho dos agentes da equipa em campo, e contém uma estrutura base que facilmente pode ser adaptada para as mais diversas aplicações de cooperação multi-agente.

Para além disso, foram estudadas duas abordagens para a fusão das bolas observadas dos diversos agentes, no supervisor. A primeira consistia na utilização de um filtro de Kalman, com o vetor de estado referenciado no referencial global. Nesta abordagem, testaram-se três hipóteses diferentes para *inputs* da fase de correção do filtro, nomeadamente, a utilização de todas as bolas observadas, com qualidade positiva, a utilização das bolas que se estivessem à distância inferior a 1m do centro de massa das observações, e a utilização de apenas a bola que tivesse maior qualidade. A segunda abordagem consistia numa heurística, que considerava que a bola do treinador era a bola que apresentasse maior qualidade. Ambas as abordagens apresentaram vantagens e desvantagens. No caso de um agente se encontrar perto da bola, constatou-se que a utilização da bola com maior qualidade apresentou melhores resultados (com ou sem filtro de Kalman). Porém, no caso de todos os agentes se encontravam à mesma distância da bola e afastados da mesma, as duas primeiras abordagens apresentaram melhores resultados. Contudo, a utilização de todas as bolas pode resultar em estimações de posição bastante desviadas da real, nomeadamente quando são observadas bolas em localizações muito dispersas. Também neste caso, a utilização das bolas que se encontrarem a menos de uma determinada distância do centro de massa das observações pode cair na situação de que nenhuma bola cumpra esse critério. Desta forma, face ao ambiente dinâmico e imprevisibilidade intrínsecos a um jogo de futebol, pode-se concluir que as abordagens que utilizam a bola de maior qualidade apresentam, em geral, melhores resultados. É ainda

de realçar que no caso de nenhuma bola estar a ser observada pelos agentes, as abordagens que utilizam um filtro de Kalman revelam uma melhor capacidade de manter uma bola válida. Na verdade, um dos principais objetivos da fusão de bola no treinador é fornecer uma estimativa de estado, mesmo quando os agentes não estão a ver a bola. Porém, esta deve ser bastante cuidadosa e ter um mecanismo de *timeout* que lhe permita detetar que a bola está perdida há muito tempo, dando-lhe oportunidade de colocar os agentes à procura da mesma.

Por fim, pode-se concluir que esta dissertação teve um contributo positivo para a equipa 5DPO, proporcionando bases sólidas que possibilitem e facilitem trabalhos futuros, fornecendo também a devida documentação.

9.2 Trabalho Futuro

Para continuação do trabalho realizado nesta dissertação, deixam-se um conjunto de sugestões para trabalho futuro.

No que concerne ao ambiente de simulação, sugere-se o desenvolvimento de um outro comportamento para a equipa adversária no simulador, adaptando o seu comportamento à situação de jogo. Por exemplo, sugere-se interligação do *software* de decisão da equipa com os adversários, permitindo que estes tenham o comportamento dos robôs da equipa. No caso do simulador correr num computador com uma placa gráfica de elevado desempenho, recomenda-se a simulação dos obstáculos a partir de objetos do tipo robô, conseguindo-se, assim, reproduzir os comportamentos de um robô real e ter duas equipas a jogar uma contra a outra.

Quanto à estimação de posição e velocidade da bola, sugere-se a implementação do filtro de Kalman com o vetor de estado referenciado no referencial local no *software* de decisão da equipa 5DPO para comparação de resultados com o filtro atualmente implementado, em situação de jogo real.

É também importante melhorar a precisão dos resultados obtidos em ambiente real. Para isso, é fundamental desenvolver um novo mecanismo de calibração de câmaras, cuja calibração seja realizada em várias orientações e não apenas numa única direção, como o sistema atualmente utilizado.

Como foi visível nos ensaios realizados, há certas situações em que se pretende obter um filtro mais rápido e outras em que se dá preferência a ter um sinal mais imune ao ruído. Como tal, sugere-se o estudo e desenvolvimento de heurísticas que façam variar o valor da covariância do erro do modelo de transição de estado Q , com base na situação de jogo.

Durante um jogo, são várias as ocasiões em que os agentes necessitam de ter uma estimativa de posição e velocidade da bola no seu referencial. É também de interesse obter uma estimativa no referencial global, principalmente para fornecer ao supervisor uma estimativa de bola que seja útil para a fusão com a dos restantes agentes. Contudo, independentemente da abordagem utilizada, ao realizar a conversão de posição e/ou velocidade entre referenciais está-se a adicionar o ruído de localização e/ou estimativa de velocidade ao estado estimado. Para obter resultados mais precisos, imunes a esse ruído e, visto que o filtro de Kalman é computacionalmente leve, sugere-se a

utilização paralela de dois filtros de Kalman, um com o vetor de estado referenciado no referencial global, e outro no referencial local, bastando-se proceder à leitura dos estados estimados, sem necessidades de conversões adicionais.

Aconselha-se a utilização de um sensor externo que permita medir a velocidade da bola, de modo a servir de *ground truth* para validação da velocidade estimada.

Por fim, é crucial a resolução do problema de odometria detetado nos ensaios realizados, sendo que este está a prejudicar gravemente o desempenho deste estimador. Sugere-se ainda a inclusão de uma IMU na geração mais antiga de robô para utilização das medidas do acelerómetro para detetar perdas de aderência das rodas, reduzindo os erros na odometria no robô e, como esta é utilizada como entrada no filtro de Kalman, reduzir o erro na estimação de posição e velocidade da bola.

Para o treinador, propõe-se a realização de uma análise comparativa da função custo atualmente desenvolvida com uma outra baseada não na distância do agente a um alvo, mas sim numa estimativa de tempo que este demora a chegar a esse alvo, ou executar uma determinada ação.

Para além disso, sugere-se uma maior utilização da tática da equipa na determinação das formações e papéis de uma determinada jogada. No caso dos agentes da equipa conseguirem distinguir obstáculos, propõe-se que estes também tenham influência na determinação de formações e papéis a utilizar.

A nível da fusão sensorial realizada no treinador, propõe-se o desenvolvimento de uma abordagem que tenha em conta a distância dos agentes às suas observações, dando um maior peso às bolas dos agentes que se encontrarem mais perto das suas observações. Porém, esta análise também deve ser feita criteriosamente, visto que um agente pode estar muito perto da sua observação e esta ser uma bola falsa. Para além disso, sugere-se a análise de uma solução híbrida que, para além de ter em conta a distância dos agentes à bola, utilize na fase de correção do filtro as bolas que estejam a menos do que uma determinada distância do centro de massa de todas as observações (com qualidade positiva) e, não havendo nenhuma bola que cumpra esse critério, utilize a bola de maior qualidade.

A nível de situações de testes, propõe-se um estudo sobre uma situação semelhante à descrita no ensaio 3, mas no caso de a bola correta ser aquela que está a ser observada pelos agentes 4 e 6.

Por fim, quando os agentes da equipa forem capazes de localizar individualmente obstáculos, sugere-se a implementação de abordagens semelhantes para a estimação da localização dos mesmos. De qualquer forma, terão de ser realizadas algumas alterações, de modo a conseguir identificar se as observações correspondem mesmo a obstáculos ou a agentes da própria equipa.

Anexo A

Ensaio do Filtro de Kalman de Estimação de Posição e Velocidade da Bola no *Software* de Decisão da Equipa 5DPO

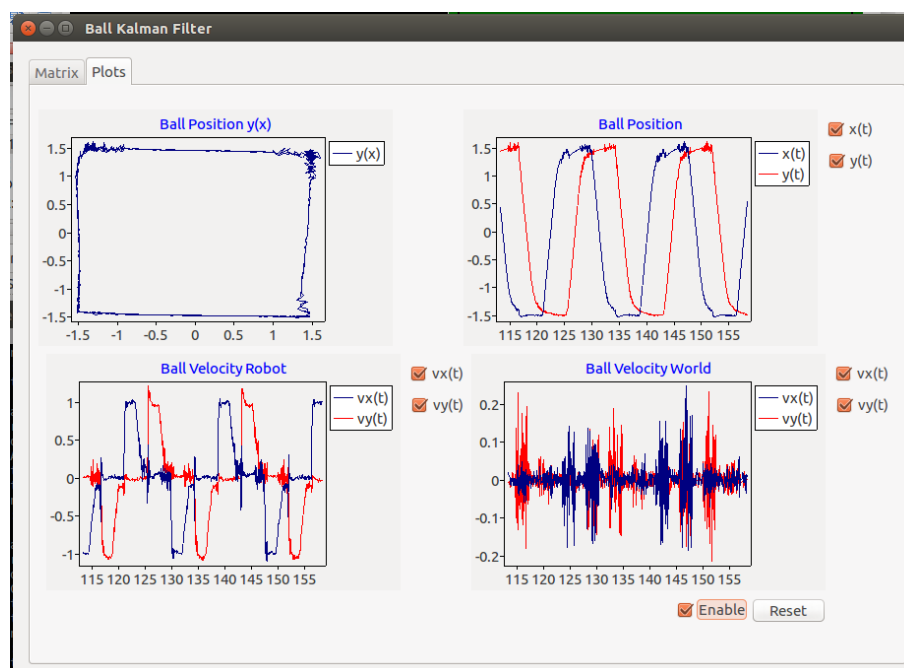


Figura A.1: Ensaio 1 do filtro de Kalman com vetor de estado no referencial global - implementação no *Decision*, com dados obtidos em SimTwo

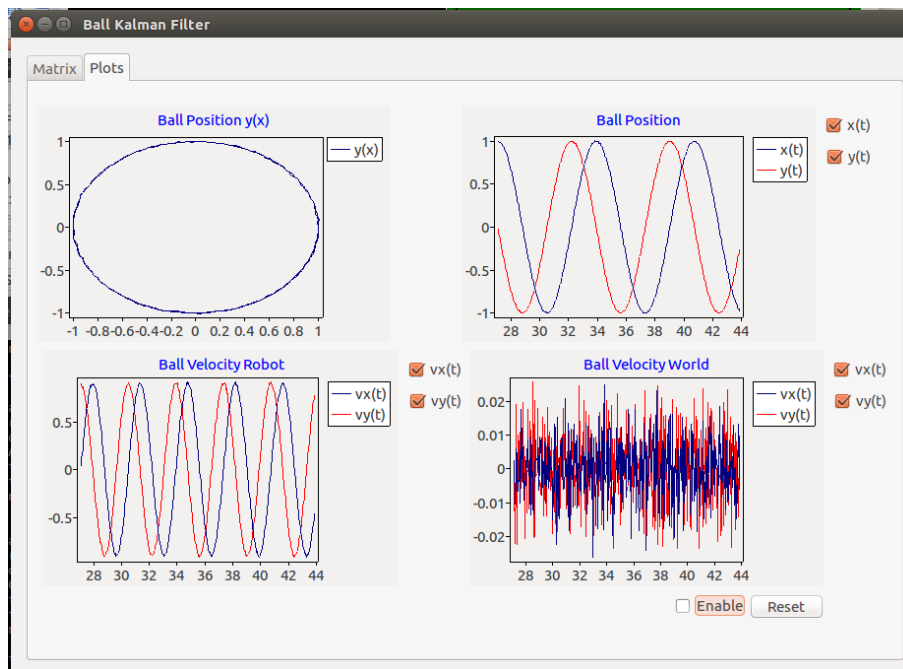


Figura A.2: Ensaio 2 do filtro de Kalman com vetor de estado no referencial global - implementação no *Decision*, com dados obtidos em SimTwo

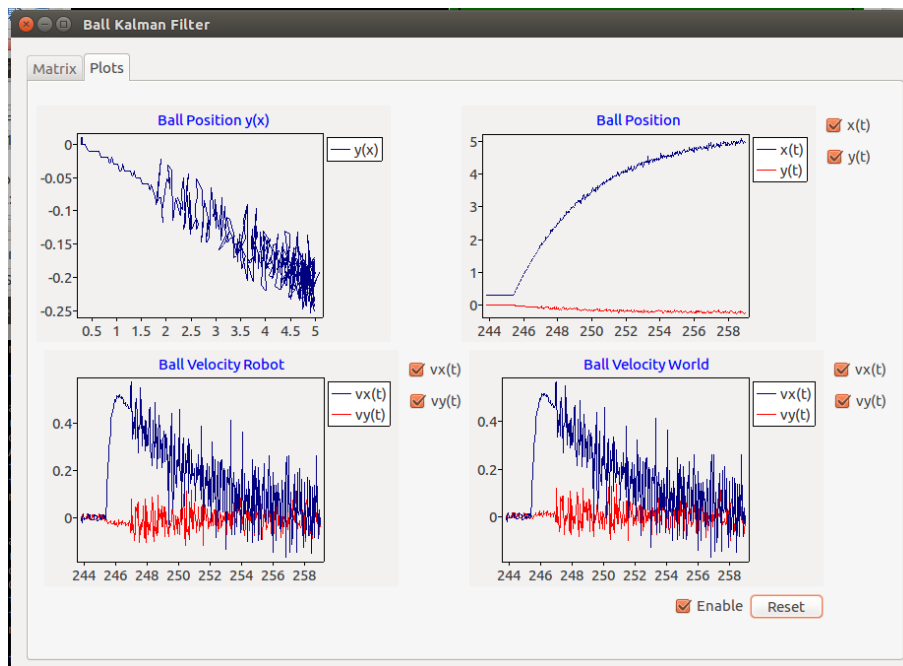


Figura A.3: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global - implementação no *Decision*, com dados obtidos em SimTwo

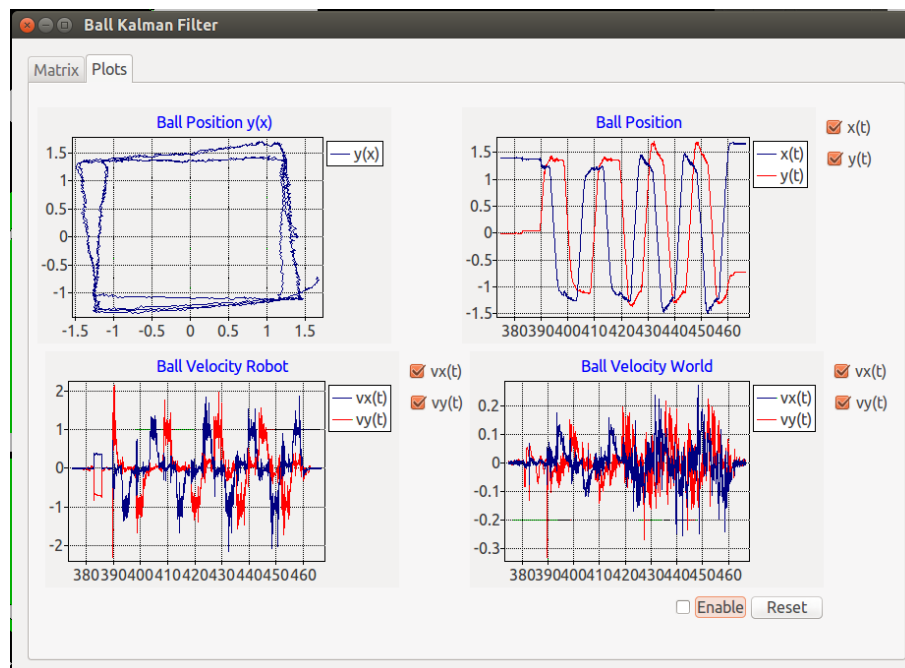


Figura A.4: Ensaio 1 do filtro de Kalman com vetor estado no referencial global - implementação no *Decision*, com dados obtidos em ambiente real

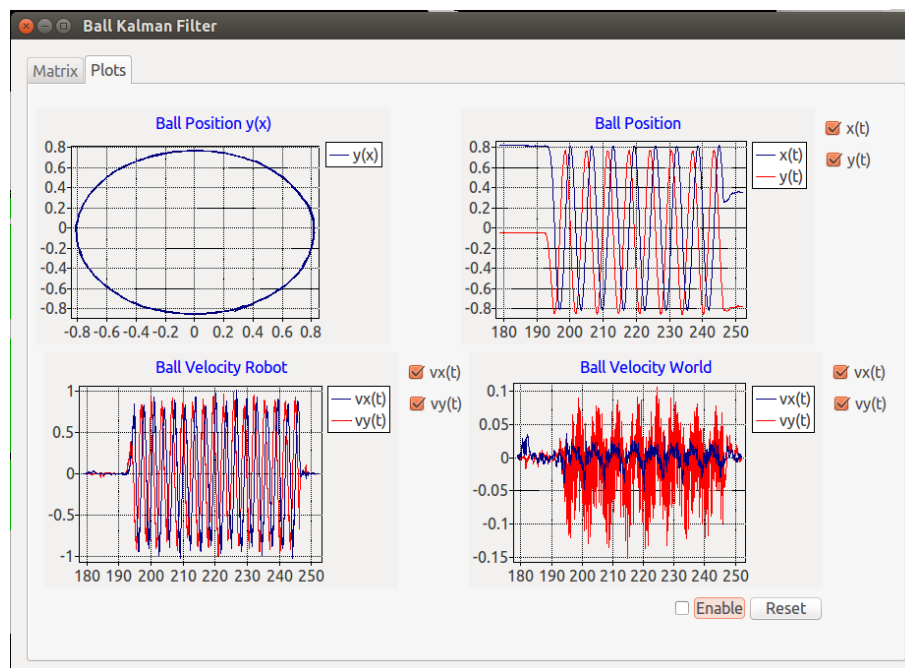


Figura A.5: Ensaio 2 do filtro de Kalman com vetor de estado no referencial global - implementação no *Decision*, com dados obtidos em ambiente real

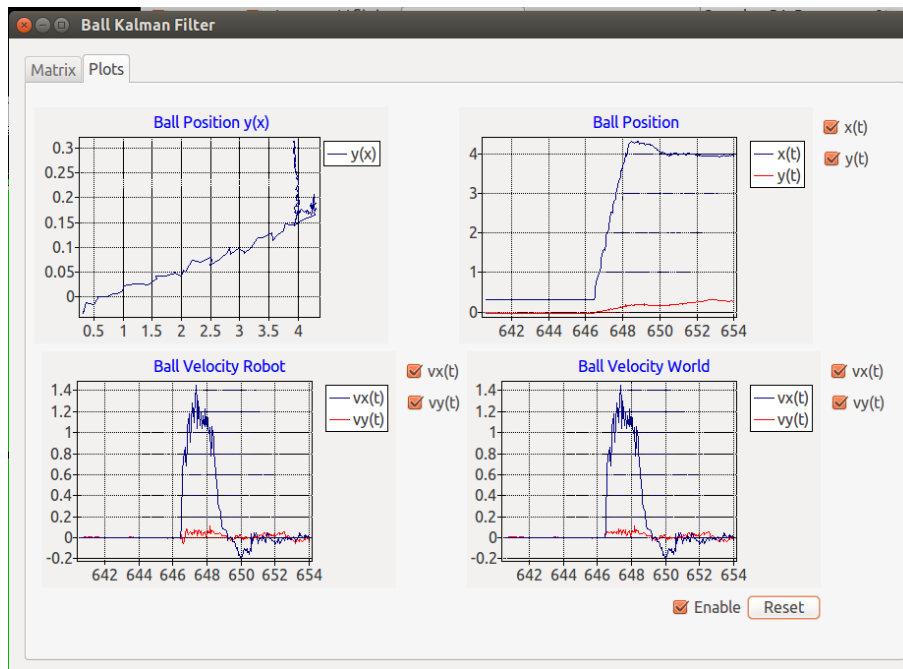
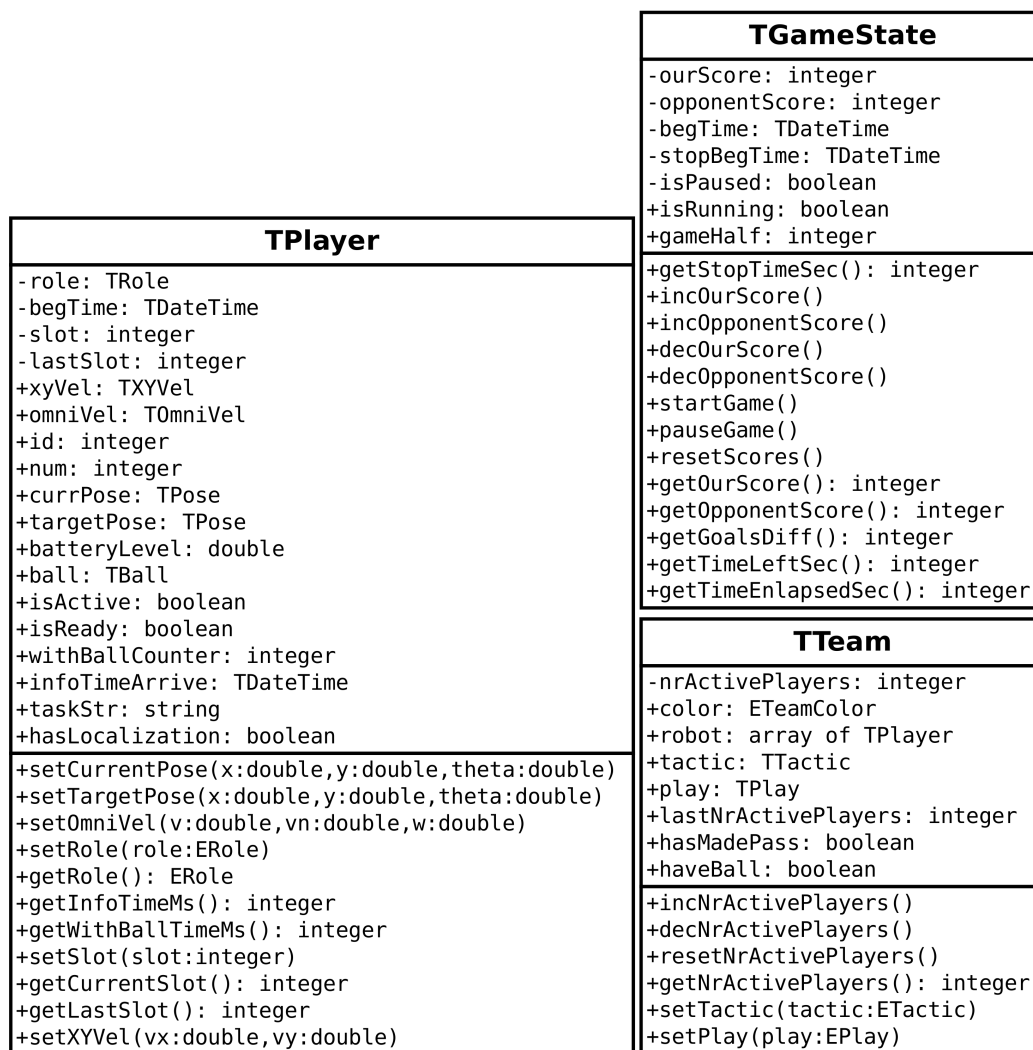


Figura A.6: Ensaio 3 do filtro de Kalman com vetor de estado no referencial global - implementação no *Decision*, com dados obtidos em ambiente real

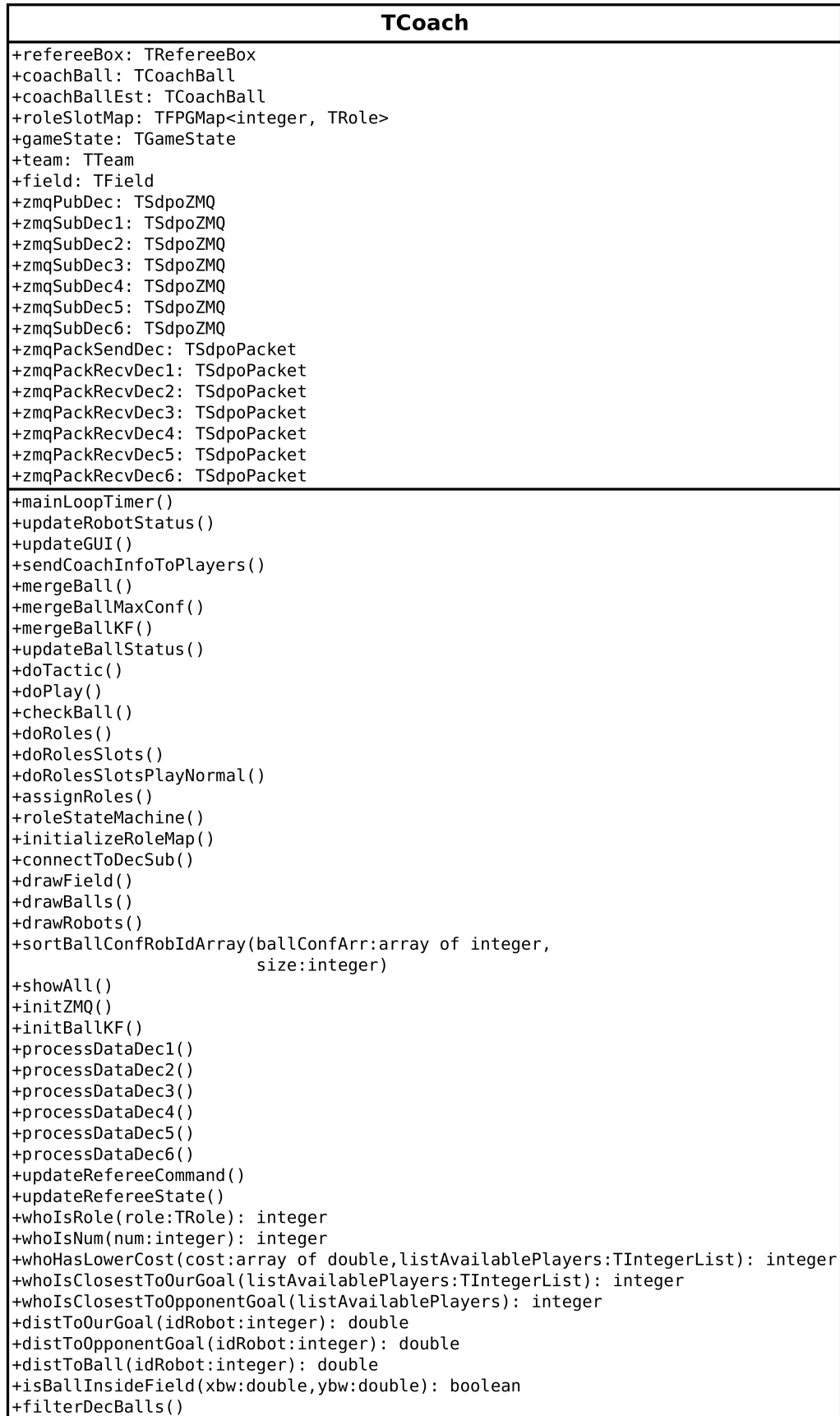
Anexo B

Treinador

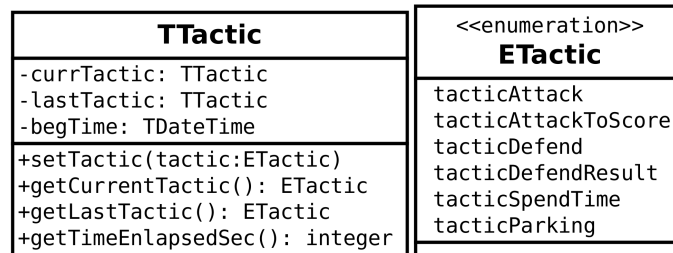
B.1 Diagrama UML das Classes e Enumerações Mais Relevantes do Treinador



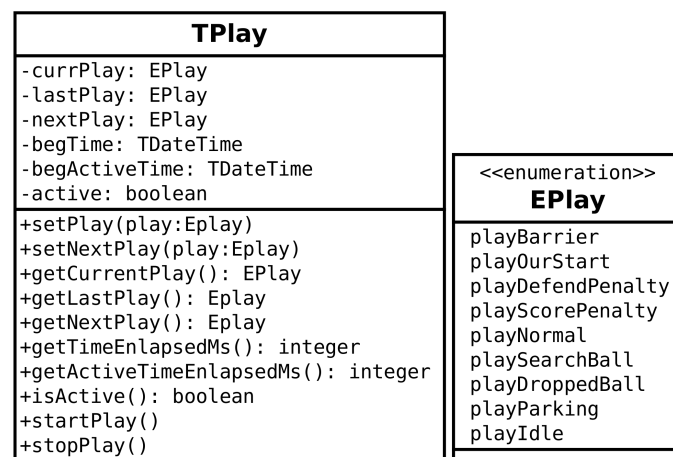
(a) Classe jogador, estado do jogo e equipa



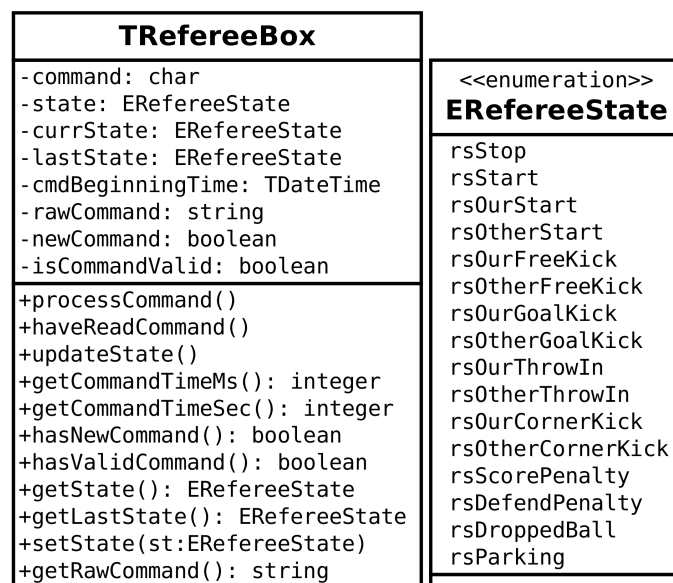
(b) Classe principal do treinador

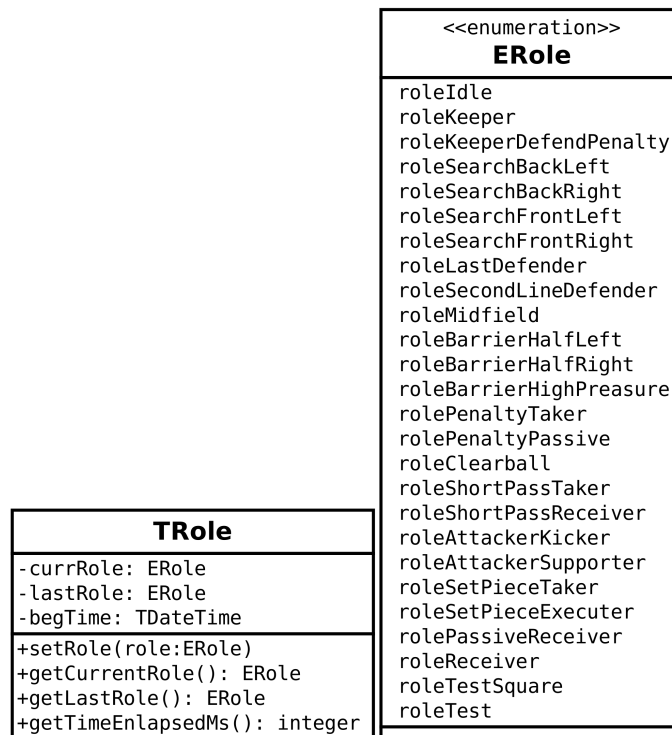


(c) Classe e enumeração de tática



(d) Classe e enumeração de jogada

(e) Classe e Enumeração do estado da *RefereeBox*



(f) Classe e enumeração dos papéis

Figura B.2: Diagrama UML das Classes e Enumerações Mais Relevantes do Treinador

B.2 Árvore de Decisão da Tática da Equipa

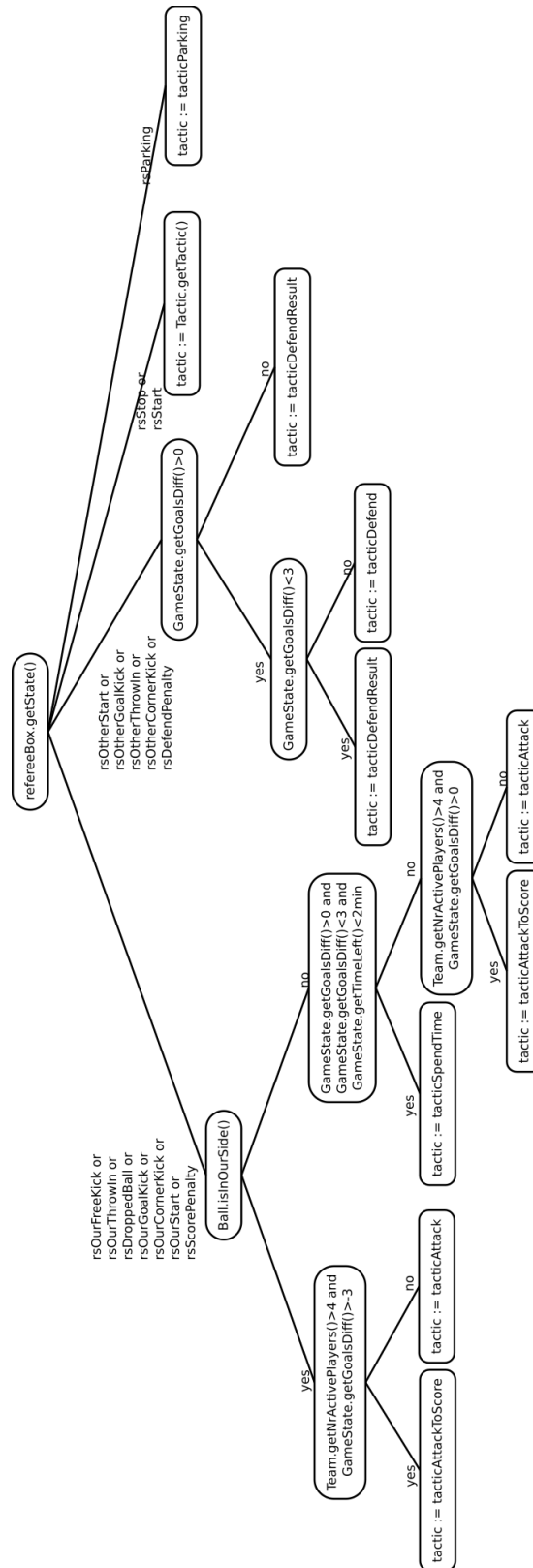


Figura B.3: Diagrama de atualização da tática da equipa

B.3 Formações e Máquinas de Estados das *Role Slots*

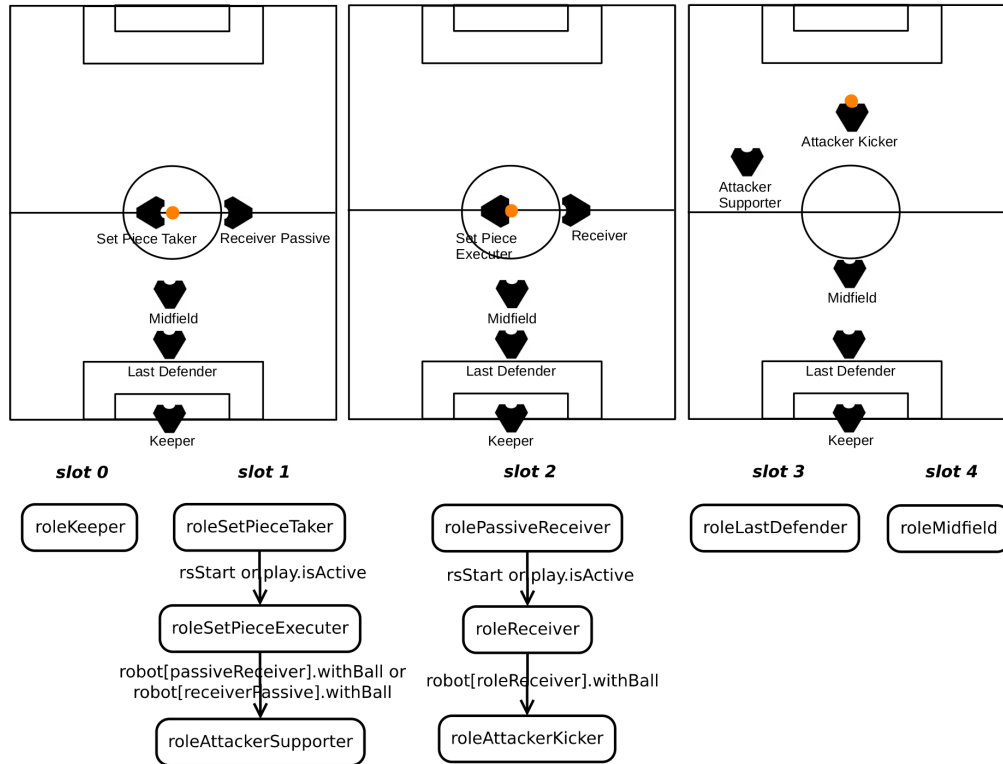


Figura B.4: Formações e máquina de estados para a jogada *playOurStart*

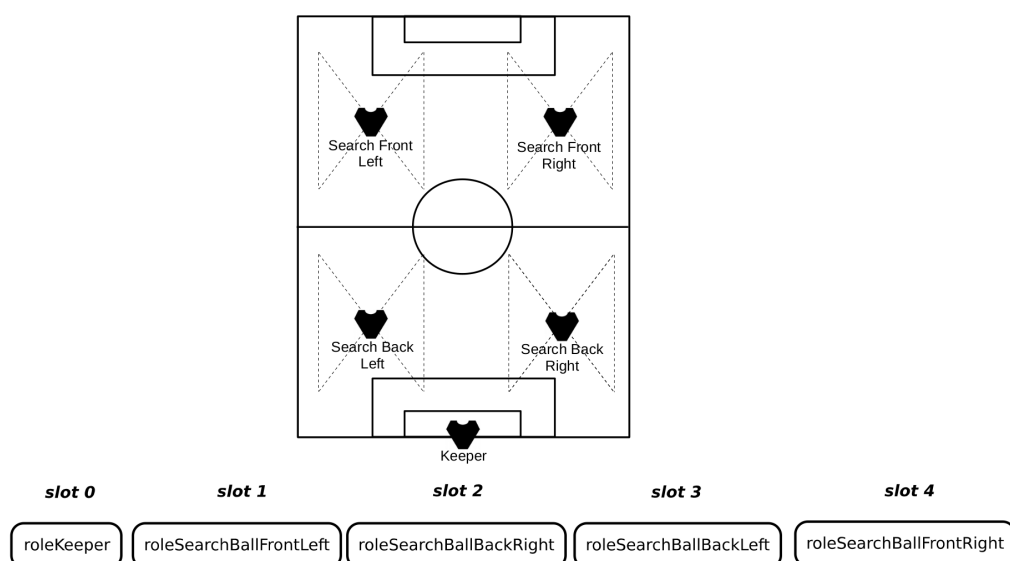
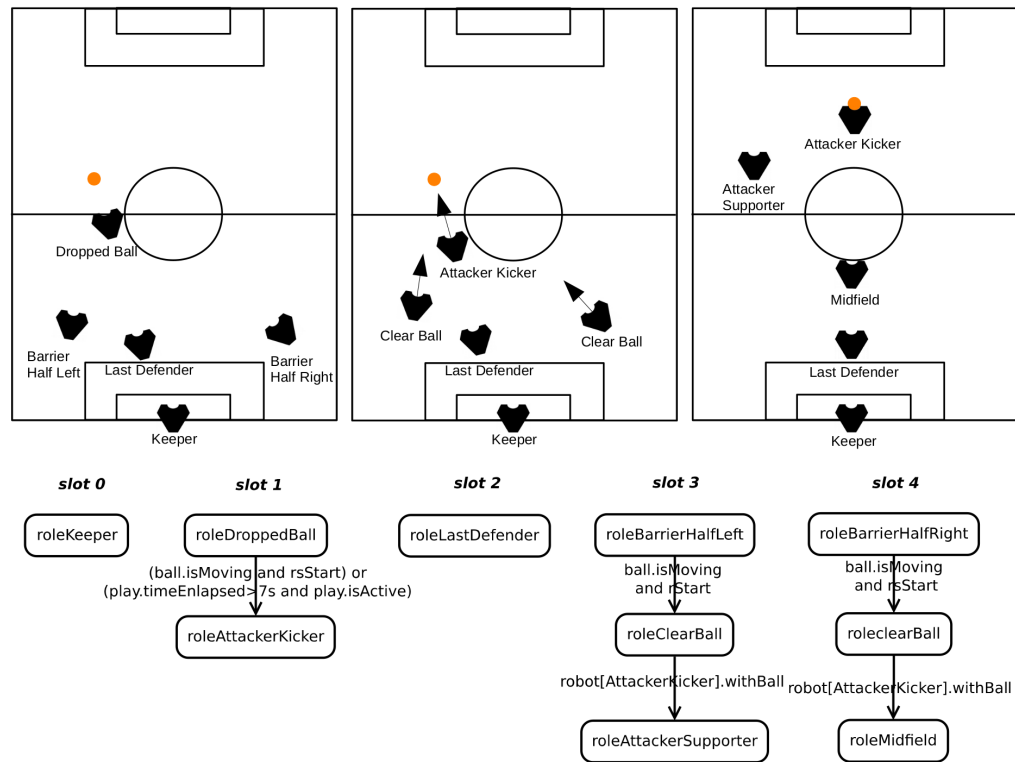
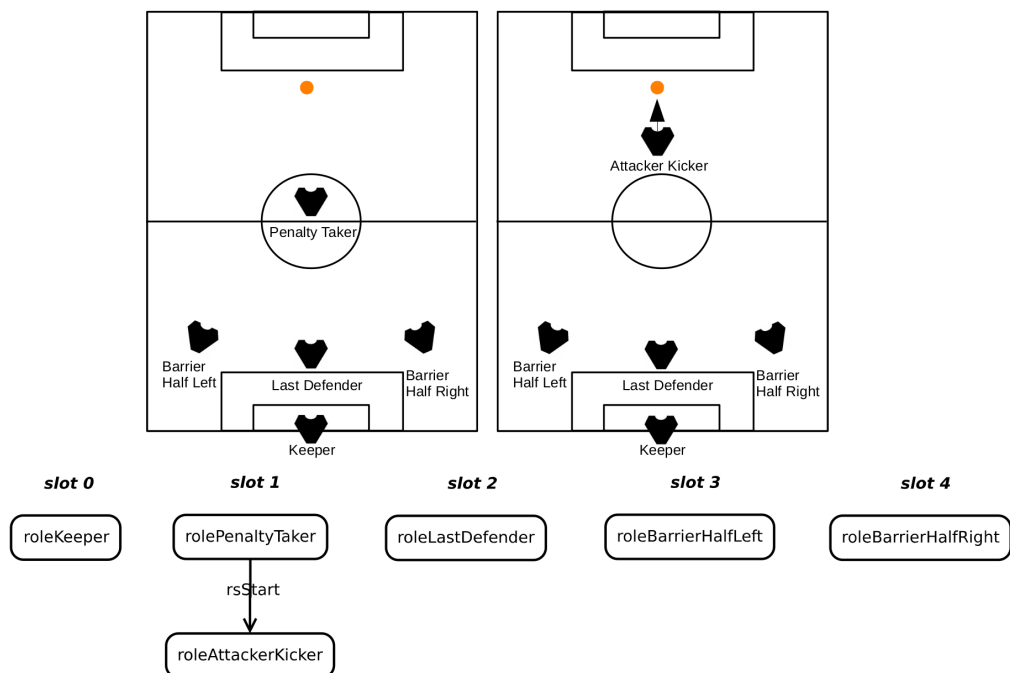
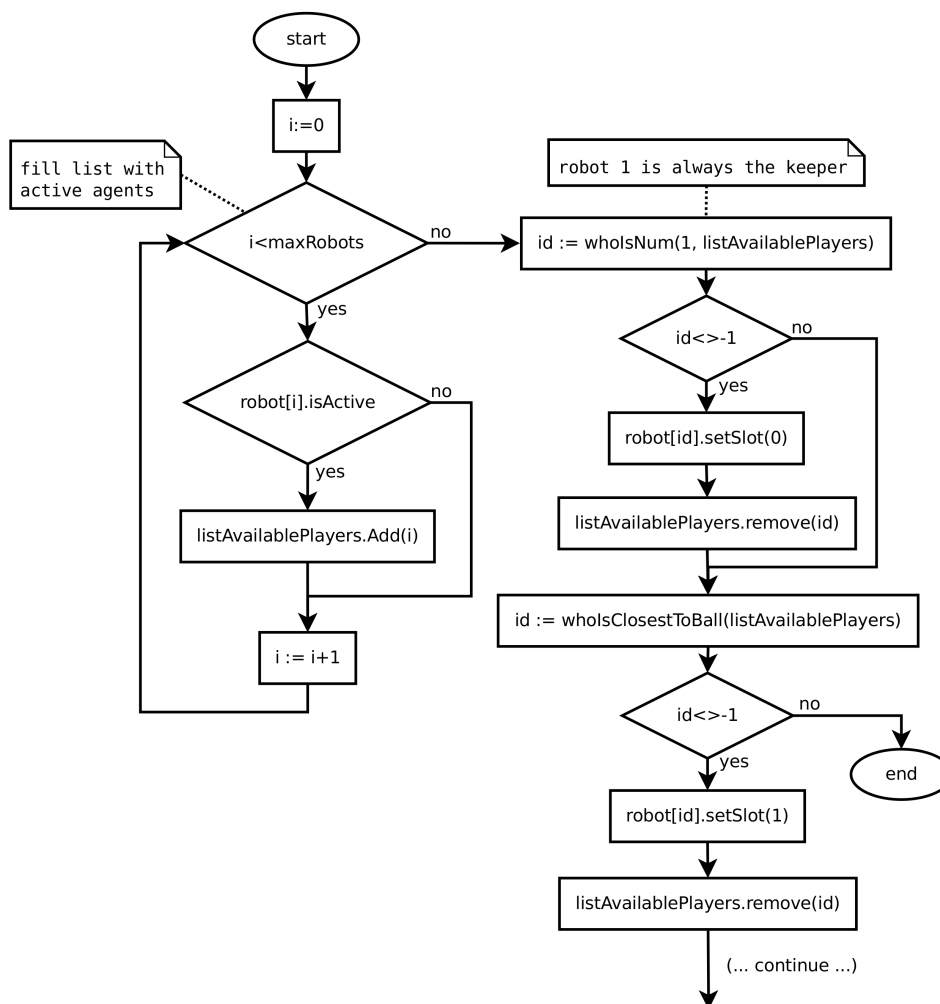


Figura B.5: Formações e máquina de estados para a jogada *playSearchBall*

Figura B.6: Formações e máquina de estados para a jogada *playDroppedBall*Figura B.7: Formações e máquina de estados para a jogada *playDefendPenalty*

B.4 Fluxograma do algoritmo atribuição de *Slots* para todas as jogadas, à exceção da *playNormal*



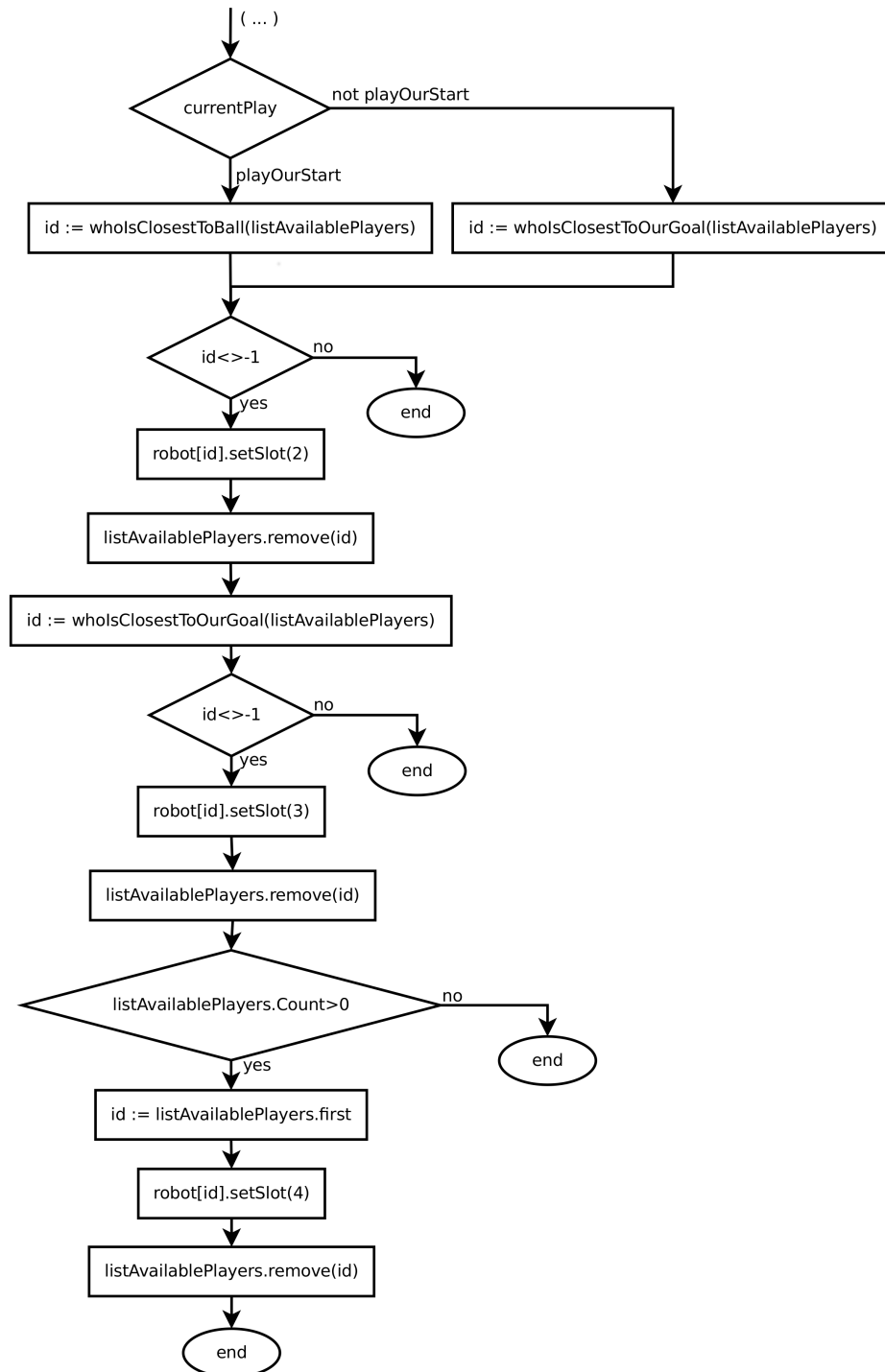
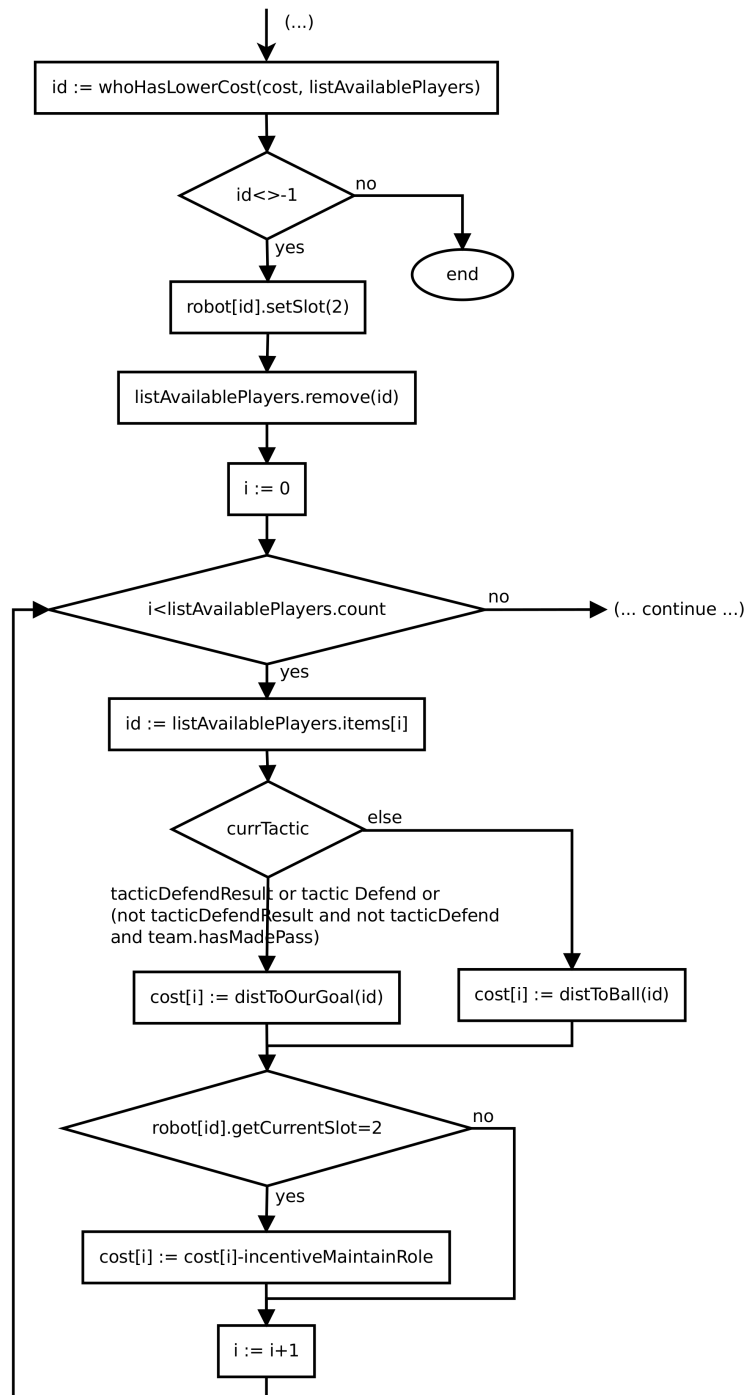


Figura B.8: Algoritmo de Atribuição de *slots* para todas as jogadas, à exceção da *playNormal*



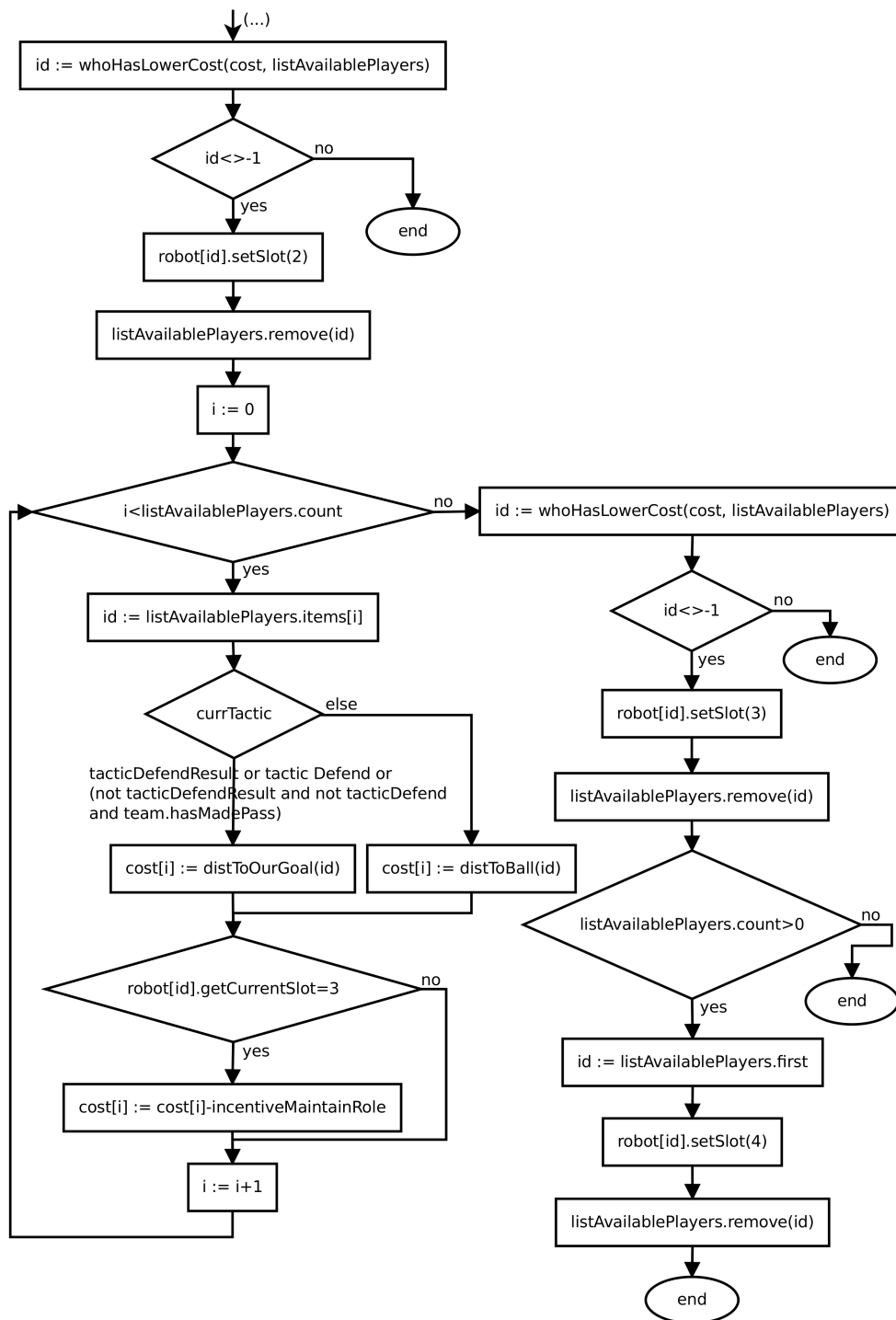


Figura B.9: Algoritmo de Atribuição de *slots* para a jogada *playNormal*

Referências

- [1] Rafael Fierro, Aveek Das, John Spletzer, Joel Esposito, Vijay Kumar, James P Ostrowski, George Pappas, Camillo J Taylor, Yerang Hur, Rajeev Alur, Insup Lee, Greg Grudic, e Ben Southall. A framework and architecture for multi-robot coordination. *The International Journal of Robotics Research*, 21(10-11):977–995, 2002. doi:10.1177/0278364902021010981.
- [2] EUA testaram com sucesso enxame coordenado de 104 drones. *Exame Informática*, página 1, jan 2017. URL: <http://exameinformatica.sapo.pt/noticias/mercados/2017-01-13-EUA-testaram-com-sucesso-enxame-coordenado-de-104-drones>.
- [3] Wolfram Burgard, Mark Moors, Cyrill Stachniss, e Frank E Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.
- [4] Rui Rocha, Jorge Dias, e Adriano Carvalho. Cooperative multi-robot systems:: A study of vision-based 3-d mapping using information theory. *Robotics and Autonomous Systems*, 53(3):282–311, 2005.
- [5] Changyun Wei, Koen V Hindriks, e Catholijn M Jonker. Dynamic task allocation for multi-robot search and retrieval tasks. *Applied Intelligence*, 45(200):383–401, 2016. URL: <http://dx.doi.org/10.1007/s10489-016-0771-5>, doi:10.1007/s10489-016-0771-5.
- [6] Jose J Acevedo, BC Arrue, Ivan Maza, e Anibal Ollero. A distributed framework for surveillance missions with aerial robots including dynamic assignment of the detected intruders. Em *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, páginas 1212–1219. IEEE, 2016.
- [7] Basilio Bona, Luca Carlone, Marina Indri, e Stefano Rosa. Supervision and monitoring of logistic spaces by a cooperative robot team: methodologies, problems, and solutions. *Intelligent Service Robotics*, 7(4):185–202, 2014.
- [8] Anibal Matos e Nuno Cruz. Coordinated operation of autonomous underwater and surface vehicles. Em *OCEANS 2007*, páginas 1–6. IEEE, 2007.
- [9] Nuno A Cruz, Bruno M Ferreira, Anibal C Matos, Chiara Petrioli, Roberto Petroccia, e Daniele Spaccini. Implementation of an underwater acoustic network using multiple heterogeneous vehicles. Em *Oceans, 2012*, páginas 1–10. IEEE, 2012.
- [10] Luís Mota, Joao A Fabro, Luis Paulo Reis, e Nuno Lau. Collaborative behavior in soccer: The setplay free software framework. Em *Robot Soccer World Cup*, páginas 709–716. Springer, 2014.
- [11] A G Paulo M Moreira. Slides de Cinemática, dinâmica e controlo de robôs. *Faculdade de Engenharia da Universidade do Porto*, 2013. URL: <https://web.fe.up.pt/>

- [~amoreira/agregacao/srau/Cap3_Cinematica_dinamica_controlo/SRAU%20Cap3%20Cinem_Dinam_Contr_v6.pdf](#).
- [12] J Z Sasiadek e P Hartana. Sensor data fusion using Kalman filter. *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, 2:WED5/19–WED5/25 vol.2, 2000. doi:10.1109/IFIC.2000.859866.
 - [13] Nadir Karam, Frederic Chausse, Romuald Aufrere, e Roland Chapuis. Cooperative Multi-Vehicle Localization. *Intelligent Vehicles Symposium*, páginas 564–570, 2006.
 - [14] Pedro U Lima, Aamir Ahmad, André Dias, André G S Conceição, António Paulo Moreira, Eduardo Silva, Luis Almeida, Luis Oliveira, e Tiago P Nascimento. Formation control driven by cooperative object tracking. *Robotics and Autonomous Systems*, 63:68–79, 2015. URL: <http://dx.doi.org/10.1016/j.robot.2014.08.018>, doi:10.1016/j.robot.2014.08.018.
 - [15] Alexander Ferrein, Lutz Hermanns, e Gerhard Lakemeyer. Comparing sensor fusion techniques for ball position estimation. Em *Robot Soccer World Cup*, páginas 154–165. Springer, 2005.
 - [16] A brief history of robocup, 2016. Disponível em http://www.robocup.org/a_brief_history_of_robocup, acessado a última vez a 25 de janeiro de 2017.
 - [17] Objective, 2016. Disponível em <http://www.robocup.org/objective>, acessado a última vez a 25 de janeiro de 2017.
 - [18] RoboCup Federation official website, 2016. Disponível em <http://www.robocup.org/>, acessado a última vez a 26 de março de 2017.
 - [19] RoboCup Wiki - Main Page, 2016. Disponível em http://wiki.robocup.org/Main_Page, acessado a última vez a 26 de março de 2017.
 - [20] Minoru Asada, Tucker Balch, Andrea Bonarini, Ansgar Bredenfeld, Steffen Gutmann, Gerhard Kraetzschmar, Pedro Lima, Emanuele Menegatti, Pieter Jonker, Alireza Fadaei Tehrani, Takayuki Nakamura, Gerald Steinbauer, Martin Lauer, Yasunori Takemura, Hui-min Lu, Enrico Pagello, Fernando Ribeiro, Thorsten Schmitt, Wei-min Shen, Hans Sprong, Shoji Suzuki, Yasutake Takahashi, Paul G Ploeger, Frank Schreiber, Akihiro Matsumoto, Saeed Shiry Ghidary, Roel Merry, Bernardo Cunha, Darwin Lau, Saeed Ebrahimijam, Oliver Zweigle, Hamed Rasam Farad, Robin Soetens, Zhao Yong, Shota Chikushi, Wu Jia Hao, Masoud Montazeri, Junhao Xiao, Ricardo Dias, Andreas Witsch, Zhao Yong, e Seyed Ehsan Marjani. *Middle Size Robot League Rules and Regulations for 2017*. 2017. URL: http://wiki.robocup.org/images/7/72/2017-msl-rules_v18_4.pdf.
 - [21] Manuela Veloso, Michael Bowling, Sorin Achim, Kwun Han, e Peter Stone. The cmunited-98 champion small-robot team. *RoboCup-98: Robot Soccer World Cup II*, páginas 77–92, 1999.
 - [22] Juan Pablo Mendoza, Joydeep Biswas, Danny Zhu, Richard Wang, Philip Cooksey, Steven Klee, e Manuela Veloso. CMDragons 2015: Coordinated offense and defense of the SSL champions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9513:106–117, 2015.

- [23] Brett Browning, James Bruce, Michael Bowling, e Manuela Veloso. Stp: Skills, tactics, and plays for multi-robot control in adversarial environments. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 219(1):33–52, 2005.
- [24] Luís Paulo Reis, Nuno Lau, e Eugénio Costa Oliveira. Situation Based Strategic Positioning for coordinating a team of homogeneous agents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2103 LNAI:175–197, 2001.
- [25] Nuno Lau e Luis Paulo Reis. Fc portugal-high-level coordination methodologies in soccer robotics. Em *Robotic Soccer*. InTech, 2007.
- [26] L P Reis, N Lau, e Springer. COACH UNILANG - A Standard Language for Coaching a (Robo)Soccer Team. *Robocup 2001 - Robot Soccer World Cup*, 2377:183–192, 2002.
- [27] M T J Spaan e F O A Groen. Team coordination among robotic soccer players. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2752:409–416, 2003. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-23144465602{&}partnerID=40{&}md5=5169a5c8d927d8c3682874780b0d633b>.
- [28] R Dias, F Amaral, J L Azevedo, B Cunha, P Dias, N Lau, A J R Neves, E Pedrosa, A Pereira, J Silva, e A Trifan. CAMBADA’2016: Team Description Paper. 2016.
- [29] N. Lau, L. SeabraLopes, G. Corrente, N. Filipe, e R. Sequeira. Robot team coordination using dynamic role and positioning assignment and role based setplays. *Mechatronics*, 21:445–54, March 2011.
- [30] António JR Neves, Filipe Amaral, Ricardo Dias, João Silva, e Nuno Lau. A new approach for dynamic strategic positioning in robocup middle-size league. Em *Portuguese Conference on Artificial Intelligence*, páginas 433–444. Springer, 2015.
- [31] Sérgio Filipe Silva Martins. Treinador automático para a equipa CAMBADA. Tese de mestrado, Universidade de Aveiro, 2013. URL: <http://hdl.handle.net/10773/12705>.
- [32] N.M. Figueiredo, A.J.R. Neves, N. Lau, A. Pereira, e G. Corrente. Control and Monitoring of a Robotic Soccer Team: The Base Station Application. Em L.S. Lopes, N. Lau, P. Mariano, e L.M. Rocha, editores, *Progress in Artificial Intelligence. Proceedings 14th Portuguese Conference on Artificial Intelligence, EPIA 2009*, páginas 299–309. Springer Verlag, 2009 2009. Progress in Artificial Intelligence. 14th Portuguese Conference on Artificial Intelligence, EPIA 2009, 12-15 Oct. 2009, Aveiro, Portugal.
- [33] R Dias, F Amaral, J L Azevedo, M Azevedo, B Cunha, P Dias, N Lau, J R Neves, E Pedrosa, A Pereira, F Pinto, D Silva, J Silva, e E Sousa. CAMBADA’2017: Team Description Paper. 2017.
- [34] Lotte de Koning, Juan Pablo Mendoza, Manuela Veloso, e René van de Molengraft. Skills, tactics and plays for decentralized multi-robot control in adversarial environments. *Submitted to AMAAS*, 2017.

- [35] Ferry Schoenmakers, Koen Meessen, Yanick Douven, Harrie van de Loo, Dennis Bruijnen, Wouter Aangenent, Bob van Ninhuijs, Matthias Briegel, Patrick van Brakel, Jordy Senden, Robin Soetens, Wouter Kuijpers, Jorrit Olthuis, Peter van Lith, Marjon van 't Klooster, Lotte de Koning, e René van de Molengraft. Tech United Eindhoven Team Description 2017. 2017. URL: <http://www.techunited.nl/media/images/Publications/TDP{ }2017.pdf>.
- [36] a.W. Stroupe, M.C. Martin, e T. Balch. Distributed sensor fusion for object position estimation by multi-robot systems. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2:1092–1098, 2001. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=932739>.
- [37] João Silva, Nuno Lau, António J R Neves, João Rodrigues, e José Luís Azevedo. World modeling on an MSL robotic soccer team. *Mechatronics*, 21(2):411–422, 2010. doi:10.1016/j.mechatronics.2010.05.011.
- [38] António J. R. Neves, José Luís Azevedo, Bernardo Cunha, Nuno Lau, João Silva, Frederico Santos, Gustavo A. Corrente, Daniel A. Martins, Nuno Figueiredo, Artur Pereira, Luís Almeida, Luís Seabra Lopes, Armando J. Pinho, João Rodrigues, e Paulo Pedreiras. CAMBADA soccer team: from robot architecture to multiagent coordination. *Robot Soccer*, páginas 19–45, 2010.
- [39] Ricardo Dias, Bernardo Cunha, Eduardo Sousa, Filipe Amaral, e Nuno Lau. Real-Time Multi-Object Tracking on Highly Dynamic Environments. Em *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, páginas 178–183, 2017.
- [40] Aamir Ahmad, Tiago Nascimento, Andre G S Conceicao, Antonio Paulo Moreira, e Pedro Lima. Perception-driven multi-robot formation control. *Proceedings - IEEE International Conference on Robotics and Automation*, páginas 1851–1856, 2013. doi:10.1109/ICRA.2013.6630821.
- [41] L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, e L.S. Lopes. Coordinating distributed autonomous agents with a real-time database: the CAMBADA project. Em C. Aykanat, T. Dayar, e I. Korpeoglu, editores, *Computer and Information Sciences - ISCIS 2004. 19th International Symposium. Proceedings (Lecture Notes in Computer Science Vol.3280)*, páginas 876–86, Berlin, Germany, 2004 2004. Springer-Verlag. Computer and Information Sciences - ISCIS 2004. 19th International Symposium. Proceedings, 27-29 Oct. 2004, Kemer-Antalya, Turkey.
- [42] Yuan Xu, Chunlu Jiang, e Yingzi Tan. SEU-3D 2006 Soccer Simulation Team Description. *CD Proc. of RoboCup Symposium 2006*, 2006.
- [43] J. Silva, N. Lau, J. Rodrigues, J.L. Azevedo, e A.J.R. Neves. Sensor and Information Fusion Applied to a Robotic Soccer Team. Em *RoboCup 2009: Robot Soccer World Cup XIII*, páginas 366–77. Springer, 2010.
- [44] Huimin Lu, Qinghua Yu, Dan Xiong, Junhao Xiao, e Zhiqiang Zheng. Object Motion Estimation Based on Hybrid Vision for Soccer Robots in 3D Space. *RoboCup 2014 International Symposium*, 2014.

- [45] Paulo José Cerqueira Gomes da Costa. *Localização em tempo real de múltiplos robots num ambiente dinâmico*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, 1999. URL: <https://repositorio-aberto.up.pt/handle/10216/11752>.
- [46] Aamir Ahmad e Pedro U Lima. Multi-Robot Cooperative Object Tracking Based on Particle Filters. *Robotics and Autonomous Systems*, páginas 37–42, 2011.
- [47] Martin Lauer, Sascha Lange, e Martin Riedmiller. Modeling moving objects in a dynamically changing robot application. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3698 LNAI:291–303, 2005.
- [48] Tech United Eindhoven Team Description 2016. 2016. URL: <http://www.techunited.nl/media/images/TDP2016.pdf>, doi:10.1007/978-3-319-18615-3.
- [49] Small size league - robocup wiki, 2017. Disponível em http://wiki.robocup.org/Small_Size_League, acessado a última vez a 22 de maio de 2017.
- [50] Edward Layer e Krzysztof Tomczyk. *Measurements, modelling and simulation of dynamic systems*. Springer Science & Business Media, 2009.
- [51] Maria Isabel Ribeiro. Kalman and Extended Kalman Filters : Concept , Derivation and Properties, 2004. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.5088{%&}rep=rep1{%&}type=pdf>, doi:10.1.1.2.5088.
- [52] Paulo Lopes dos Santos. Notas de Aula sobre Filtro de kalman, 2012. URL: https://sigarra.up.pt/feup/pt/conteudos_service.conteudos_cont?pct_id=301512&pv_cod=56CYiaDSk1E1.
- [53] E.a. a Wan e R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *Technology*, v:153–158, 2000. URL: http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=882463, doi:10.1109/ASSPCC.2000.882463.
- [54] Simon J Julier e Jeffrey K Uhlmann. New extension of the Kalman filter to nonlinear systems. *Int Symp AerospaceDefense Sensing Simul and Controls*, 3:182, 1997. URL: <http://link.aip.org/link/?PSI/3068/182/1{%&}Agg=doi{%&}5Cnhttp://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.280797>, arXiv:arXiv:1011.1669v3, doi:10.1117/12.280797.
- [55] Welcome to the 5dpo site, 2011. Disponível em <https://web.fe.up.pt/~robosoc/en/doku.php>, acessado a última vez a 26 de abril de 2017.
- [56] Pedro Luís Cerqueira Gomes da Costa. *Planeamento Cooperativo de Tarefas e Trajetórias em Múltiplos Robôs*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, 2011. URL: <https://repositorio-aberto.up.pt/handle/10216/62107>.
- [57] André Scolari Conceição, A Paulo Moreira, Paulo J Costa, e Luis Paulo Reis. Architecture of cooperation for multi-robot systems. *IFAC Proceedings Volumes*, 39(20):45–50, 2006.
- [58] Paulo Costa, J Gonçalves, J Lima, e P Malheiros. Simtwo realistic simulator: A tool for the development and validation of robot software. *Theory and Applications of Mathematics &*

- Computer Science*, 1(April):17–33, 2011. URL: <http://bibliotecadigital.ipb.pt/handle/10198/4117>.
- [59] paco/wiki | simtwo, 2015. Disponível em <http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo>, acessado a última vez a 28 de maio de 2017.
- [60] Renato Miguel Dos Santos Caldas. *Modelação e Simulação de um Robot Omnidirecional de 3 rodas*. Tese de mestrado, Universidade do Porto, 2009. URL: <https://repositorio-aberto.up.pt/handle/10216/59099>.
- [61] André Gustavo Conceição, António Moreira, Paulo Costa, Pedro Costa, e Tiago Nascimento. Modeling omnidirectional mobile robots: An approach using simtwo. Em *CONTROL'2012*, 2012.
- [62] Tiago Pereira do Nascimento. *Coordinated multi-robot formation control*. Tese de doutoramento, Universidade do Porto, 2012. URL: <https://repositorio-aberto.up.pt/handle/10216/72678>.
- [63] J Hierrezuelo e C Carnero. Sliding and rolling: the physics of a rolling ball. *Physics Education*, 30(3):177, 1995.
- [64] Armando Jorge Miranda de Sousa, Paulo José Cerqueira Gomes da Costa, e António Paulo Gomes Mendes Moreira. Sistema de localização de robôs móveis baseado em filtro de kalman estendido. Em *Proceedings of the Scientific Meeting of the Portuguese Robotics Open*, volume 1, páginas 83–88, 2004.
- [65] T Soler e M Chin. On transformation of covariance matrices between local cartesian coordinate systems and commutative diagrams. Em *ASP-ACSM Convention*, páginas 393–406, 1985.
- [66] Kirk Wolter. *Introduction to variance estimation*. Springer Science & Business Media, 2007.
- [67] Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Giorgio Grisetti, e Kai Aras. Introduction to Mobile Robotics - Error Propagation Slides, 2010. URL: <http://ais.informatik.uni-freiburg.de/teaching/ss10/robotics/slides/11-error-propagation.pdf>.
- [68] Martin Lauer, Sascha Lange, e Martin Riedmiller. Calculating the perfect match: an efficient and accurate approach for robot self-localization, robocup 2005: robot soccer world cup ix, 2006.
- [69] Douglas C Montgomery e George C Runger. *Applied statistics and probability for engineers*. John Wiley & Sons, 2010.
- [70] Distributed systems are in out blood, 2017. Disponível em <http://www.imatix.com/>, acessado a última vez a 15 de abril de 2017.
- [71] Ømq language bindings, 2017. Disponível em http://zeromq.org/bindings:_start/p/1, acessado a última vez a 15 de abril de 2017.
- [72] P. Hintjens. *ZeroMQ: Messaging for Many Applications*. Oreilly and Associate Series. O'Reilly Media, Incorporated, 2013. URL: <https://books.google.pt/books?id=KWtT5CJc6FYC>.

- [73] A Dworak, P Charrue, F Ehm, W Sliwinski, e M Sobczak. Middleware Trends And Market Leaders 2011. *13th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2011)*, C111010:4, 2011. URL: <http://cds.cern.ch/record/1391410>.
- [74] A Dworak, F Ehm, P Charrue, e W Sliwinski. The new CERN Controls Middleware. *Journal of Physics: Conference Series*, 396(1):012017, 2012. URL: [http://iopscience.iop.org/1742-6596/396/1/012017{%}5Cnhttp://stacks.iop.org/1742-6596/396/i=1/a=012017?key=crossref.23ff21fa2ab23260b428f8d53ecb626b](http://iopscience.iop.org/1742-6596/396/1/012017/%5Cnhttp://stacks.iop.org/1742-6596/396/i=1/a=012017?key=crossref.23ff21fa2ab23260b428f8d53ecb626b), doi:10.1088/1742-6596/396/1/012017.
- [75] Ømq learn the basics, 2017. Disponível em <http://zeromq.org/intro:read-the-manual>, acessado a última vez a 14 de abril de 2017.